Vibration Reduction in a Gantry Robot

Marcus Peterson

Department of Automatic Control

MSc Thesis
TFRT-6075
ISSN 0280-5316

Department of Automatic Control
Lund University
Box 118
SE-221 00 LUND
Sweden

Printed in Sweden by Tryckeriet i E-huset
Lund 2018

Abstract

This report considers the problems of using a model-based approach to develop an estimation for a non-linear system using a linear model. Different estimation approaches were researched and evaluated using an ideal system, thus finding the suitable method for this thesis. The investigations considered both drive-train friction and the identification of oscillatory modes in a real industrial robot system of gantry-type and in a couple of laboratory setups, with different levels of achieved accuracy and reproducibility. A simple experiment with model-based reference compensation from identified oscillatory dynamics motivates the estimation problem and illustrate achievable benefits.

Contents

1

Introduction

The field of robotics is rapidly expanding and a lot of different actors on the market are today trying to supply control platforms to control robots. However, many of these platforms are on a high level of abstraction, meaning little to no possibility of low level configuration of robotic movement. An interesting question to pose is therefore whether low level control systems such as Beckoff TwinCat [1] or or e.g., Siemens motion control or PLC-control systems [2] can be used to control robotics in an arbitrarily sufficient way. A problem that in particular needs to be analyzed is whether a robot arm with a fixed tool can be controlled with good precision with regards to the tool rather than the joints, thus reducing the risk that the tool collides with other objects in the workspace due to oscillations on the arm side.

During motion vibrations can occur due to the weight of the robot load , or "cart", causing movement in the whole system. This gives rise to vibrational positional offsets in the tool which are poorly damped by simple regulators for controlling the angle or position of the motor side of the robot joints. Thus a control strategy must be developed such as to identify these vibrations and their origin, in order to be able to eliminate them.

1.1 Background

The interest for the field of robotics is rapidly increasing. This is due to the overwhelming benefits that can be achieved through automation of industries, which can both increase efficiency as well as safety.

However, going from the experienced and careful touch of an experienced human to the logical reasoning of a machine is not a straight path. It requires much thought towards design and purpose of a machine, as well as heavy mathematical calculations, so as to reach a higher efficiency and precision

[1] `www.beckhoff.com/twincat` Visited: 051218

[2] https://www.siemens.com/global/en/home/products/automation/systems.html Visited: 051218

than that of a human. It is therefore proper to take into account what kind of work one is trying to automate. Is it a simple movement from point A to point B, or does it require judgment calls from the operator? Will the operation require fine tuned precision or are the scales of the problems so large that a small error might not have a big impact? A simple movement might only require some form of PID control algorithms to achieve satisfactory results. It is commonly believed that in industry, around 85-95% of control loops are closed with variants of the PID control algorithm [**8**], and they are not seldom poorly tuned, by just setting the different values to unity. For most purposes this is indeed not an issue. The residual error might be small enough to be offset by a human operator, or the slow rise time of the PI-controller might not be an issue in some parts of the factory. These situations require little to no modeling and will achieve the goals set up by the management despite these issues. However, when working in a high-paced and precise environment, these issues can create large problems in efficiency and wear and tear. This is where a mathematical and automatic control approach can supply a way to reduce this problem.

1.2 Scope of thesis

In this thesis, the object was at first to analyze many different control platforms and analyze them on the grounds of their performance as well as their availability regarding incorporation of different sensors and actuators. The process through which this would be tested was a dual motor cart on a rack and pinion, located in the robotics lab of Lund University. Due to delivery times of certain parts however, big delays were introduced into the work of the thesis. At a late date in the work, a contact was established with the company Güdel in Switzerland, and their problem with a Gantry robot.

The robot had unwanted vibrations in operation, and the company had a clear goal in sight, which was to eliminate these vibrations and thus achieve higher efficiency in production, as well as reduce wear and tear on the robot itself.

Thus the object of the thesis was changed from the analysis of platforms, to investigating this process, and try to eliminate the vibrations. The aim was thus to model the process, verify and identify it, apply that model to a Kalman Observer, and then implement controllers so as to control around it.

The big questions are then: are linear approximations of non-linear processes valid and useful for a process of this size? What is a suitable control strategy for this kind of process? How can observed resonant frequencies be handled within a control loop?

A non-linear dynamic within a process is something that is often present in real-world processes, mainly in the form of saturations as well as hysteresis

in movement due to friction, and backlash in actuator drive-chains and gearboxes. These are troubling when a fine precision is needed, and have been the subject of much study in recent years, due to the ever smaller dimensions of parts productions. Some of these effects must be taken into account and can not be worked around, such as backlash and saturation. A backlash-free gearbox will not work due to thermal expansion, and a motor will always have an upper limit of its output.

Satisfactory control might even still be achieved if these things are kept in mind when designing the control strategy. The question is rather, how do we design our control so as to not be troubled by these phenomena?

Friction is however something that is always present and must be taken into account. In an ideal world, a motor would only need a short torque impulse to keep running forever at desired speed. However, friction causes it to come to a halt eventually. Thus it must be taken into account.

1.3 Outline of thesis

What is then the best way to approach such an investigation of such a process? A proper first step is to think about how to model such a process. This is in itself a big challenge and can be fraught with challenges, since real world processes do not always lend themselves easily to mathematical modeling. What kind of disturbances are acting on the process? Are the dynamics linear or non-linear? If non-linear, how well can it be approximated by a linear model? Can assumptions about behavior be made in such a way that our model becomes simple and still contains real world insights and effects?

These questions must be analyzed and thought of, so as to have a solid foundation to stand on, when working towards a good control of the process. After the initial step of establishing a model, a verification of this process must be made. In essence, this is tantamount to identifying the system, since there might be unknown constants in our systems, due to the way the system has been modeled. A number of questions relating to identifying are then posed. How are we collecting the data from the system? How are those data handled? Are there connections and couplings in the data that might cause us to identify the wrong thing when trying to identify the system? A common problem is trying to identify part of a control loop, but if that loop is closed, one might accidentally identify the inverse of the rest of the system instead, depending on how strongly the data is correlated. Are we overfitting the model so as to reach accuracy locally at the expense of global inaccuracy?

Once these questions have been properly answered and thought of, the next step can be taken, which is to decide upon a control strategy for the system.

2

Theory

2.1 Modeling of setup

A big problem in all mathematical approaches to real-world problems is where to draw the line in precision. All real world problems come with a multitude of real world effects that have an impact on the system, and that can change the dynamics or behavior of it. A few questions to ask are then:

- What kind of disturbances and dynamics are there in the system?
- Do these aforementioned disturbances and dynamics have a large scale impact on the system?
- Are the disturbances close to the working range of the system?
- How high of an order does the system have, and can it perhaps be approximated by a lower order system?
- Are there such situations that the system might come close to exciting the disturbance frequencies during operation?
- What information can we extract from the system in a meaningful way so as to help us build our model and verify it?

These questions have to be thought of and answered so as to know where to start modeling and approximating a system. It is important to know, that no mathematical model can reach 100% real-world accuracy, rather, models are always approximations. After a model has been proposed, it must be verified to fit the empirical data collected from the system. Then another question of interest is raised, and that is the question of validation. It may seem straightforward at first to just take the data and compare it to the expected data from the model and to just try to fit the model to the data, so as to achieve higher accuracy. This may cause problems when operating in ranges outside of the ones in the collected data, since the model might have only been fitted to the actual range. This is called *overfitting*.

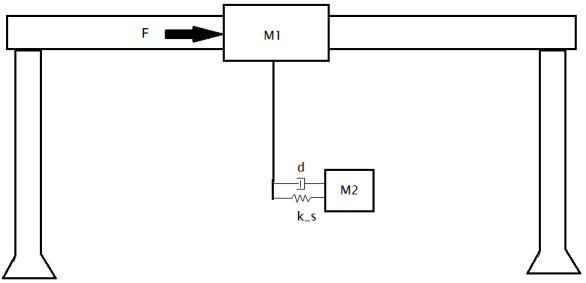


Figure 2.1 Model of the system with elasticity taken into account. $M1$ is the mass of CMP1, $M2$ is the mass of CMP2, F the input force, k_s the spring constant and d the dampening constant

To be able to dynamically model the gantry system considered in this thesis project we divide it into two main focus points; the position of the cart, p_1, and the position of the tool, p_2. These two positions are assumed to be the center of mass points, here-after known as CMPs. The system was modeled using a linear equation system for the two masses. The cart is assumed to be connected to an inflexible rod reaching down to the lower mass. The lower mass is connected to the rod via a spring and a dampener, which can be seen in Figure 2.1. The only input to the system is thus the torque to the top mass, which is counteracted by the viscous friction, which is proportional to the velocity. Since the momentum of the moving cart, beam and load are large, one must consider the elasticity and bending of the materials used to construct the system, as seen in Figure 2.1.

Thus, the connection between the two CMPs is modeled as two masses being connected via a spring and a dampener, with a force F acting upon the upper CMP.

Since the system is modeled in this way we completely ignore the gravitational component, since the only dislocations that exist are in the y-direction of the system in this idealized model.

The correctness of this approach can be discussed, since the flexion of the beam is non-linear, and since the beam is of fixed length, any bending will result in a decrease of the z-position. However, since the bending of the beam is assumed to be small, we can assume that the z-position will

not be influenced in such a way that the gravitational component will be of substantial influence to the motion of the system. The equations then become, with p being the position of either center mass point, $\dot{p}$ their corresponding velocities and $\ddot{p}$ the accelerations:

$$m_1\ddot{p}_1 = F - k_s(p_1 - p_2) - d(\dot{p}_1 - \dot{p}_2) \tag{2.1}$$

$$m_2\ddot{p}_2 = k_s(p_1 - p_2) - d(\dot{p}_2 - \dot{p}_1) \tag{2.2}$$

However, the spring and dampener variables can not be assumed to be constant due to the flexion of the bearing beam. They are dependent on the position of the cart, since the flexion increases with distance to the supports. Thus, the equations become:

$$m_1\ddot{p}_1 = F - k_s(p_1)(p_1 - p_2) - d(p_1)(\dot{p}_1 - \dot{p}_2) \tag{2.3}$$

$$m_2\ddot{p}_2 = k_s(p_1)(p_1 - p_2) - d(p_1)(\dot{p}_2 - \dot{p}_1) \tag{2.4}$$

The force acting upon p_1 is modeled as

$$F = F_{motor} - F_{friction} \tag{2.5}$$

The system matrices hence become

$$\begin{bmatrix} \dot{p}_1 \\ \ddot{p}_1 \\ \dot{p}_2 \\ \ddot{p}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_s(p_1)}{m_1} & -\frac{d(p_1)}{m_1} & \frac{k_s(p_1)}{m_1} & \frac{d(p_1)}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{k_s(p_1)}{m_2} & \frac{d(p_1)}{m_2} & -\frac{k_s(p_1)}{m_2} & -\frac{d(p_1)}{m_2} \end{bmatrix} \begin{bmatrix} p_1 \\ \dot{p}_1 \\ p_2 \\ \dot{p}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} F \tag{2.6}$$

$$[y] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} [p] \tag{2.7}$$

2.2 Friction

An inherent part of any moving system is friction. It arises from the contact of any two surfaces as a consequence of the forces between them. To be able to model it, one must first try to understand the underlying causes of it, and how they affect the system. Any system that moves on a surface is counteracted upon by friction. Whereas only viscous friction was included in the system model of Eq. (6), we here describe and consider several alternative friction models. Often, friction is simply modeled by a static force, anti-parallel to the movement vector along the surface, as the weight of the moving system times a friction constant.

$$F_{fr} = \mu N \tag{2.8}$$

It is a very rudimentary model and in many cases it is sufficient for surface-to-surface contact in a normal setting. The only parameter to consider is μ, which is a constant that is decided by which materials are involved in the movement. A box sliding on ice has a lower constant than a box being pushed along an asphalt road. The most underlying principle behind friction is the forming of bonds between two surfaces touching one another, such as Van der Waals[1] or hydrogen bonding[2]. These are weak bonds compared to the other types of electric bonds found in nature. This was further developed by Coulomb to become the formula, with v as speed

$$F_{fr} = F_C \operatorname{sgn}(v) \tag{2.9}$$

which is called the Coulomb friction.

It is also worth to consider that this force does not reflect reality, as friction can not make an object move by itself, it is merely a counteracting force. Therefore, in modeling, it must be considered as a rising force when displacement is zero, reaching a maximum when displacement starts to occur.

$$F_{fr} = \begin{cases} F & \text{if } v = 0 \text{ and } |F| < F_S \\ F_S \operatorname{sgn}(F) & \text{if } v = 0 \text{ and } |F| \geq F_S \end{cases}$$

where F_S is the static friction force. Thus we have a friction model that is based not on velocity, but on external force. This is what is called static friction, or stiction for short.

An ubiquitous way of reducing friction is to introduce some form of lubricant between the two surfaces. The resulting friction is derived from the theory of hydrodynamics, with F_v as a viscous friction force and was first modeled as

$$F_{fr} = F_v v \tag{2.10}$$

When the two surfaces are in motion in regards to one another. However, this was shown not to be sufficiently accurate due to its linear nature. A more common model for viscous friction is

$$F_{fr} = F_v |v|^{\delta_v} \operatorname{sgn}(v) \tag{2.11}$$

where the parameter δ_v depends on the geometry of the surfaces.

The above equations are then put together to try to create a continuous describing function of the frictional force in the different working ranges. The equation becomes

$$F_{fr} = \begin{cases} F_{fr}(v) & \text{if } v \neq 0 \\ F & \text{if } v = 0 \text{ and } |F| < F_S \\ F_S \operatorname{sgn}(F) & \textit{otherwise} \end{cases}$$

[1] `https://en.wikipedia.org/wiki/Van_der_Waals_force`
[2] `https://en.wikipedia.org/wiki/Hydrogen_bond`

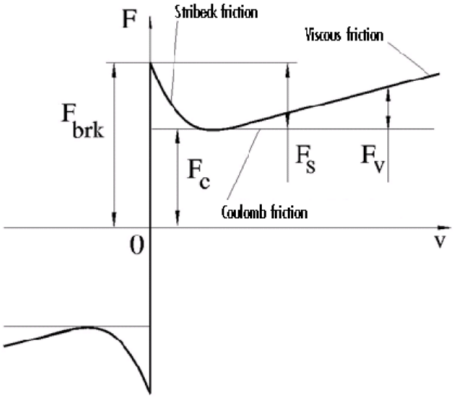


Figure 2.2 The force velocity relationship on friction.F_{brk} is the breakaway force, i.e. the force needed to break away from stationarity. Note that the model doest not need to be symmetric, and can have different values of the parameters in the different directions of force/velocity. [*https* : *//se.mathworks.com/help/physmod/simscape/ref/translationalfriction.html*].

The dynamic equation of the friction force is something that has been modeled in a lot of different ways. One parametrization of it which is the most common is found below[**5**][**6**].

$$F_{fr} = F_C + (F_S - F_C)e^{-|v/v_s|^{\delta_s}} + F_v v \tag{2.12}$$

where v_s is the Stribeck velocity. The δ_s term is a design parameter to be suited to the different applications used. In this work, due to the constant application of lubrication from the system, the parametrization as seen below was chosen to model the system

$$F_{fr} = (F_C + (F_S - F_C)e^{-|v/v_s|^{\delta_s}} + F_v|v|^{\delta_v})\ \mathrm{sgn}(v) \tag{2.13}$$

This was the final parametrization used to model the friction between the cart and the beam on the robot, and has a force velocity relationship as seen in figure 2.2.

2.3 Model representations - Continuous and discrete Time

Choosing an appropriate sampling period is rather more complex than just choosing an arbitrary number. With a shorter sampling period comes an increased load on the computational efficiency, and with a longer one important behaviors of the system might go by unnoticed [**2**].

So how do we go from continuous representation to discrete? Since we know that a good representation of a continuous linear system is

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{2.14}$$
$$y = Cx(t) + Du(t) \tag{2.15}$$

we would like to find a similar representation that only changes at set intervals, which means we want to find matrices Φ and Γ, so that we get

$$x(t_{k+1}) = \Phi(t_{k+1}, t_k)x(t_k) + \Gamma(t_{k+1}, t_k)u(t_k) \tag{2.16}$$
$$y(t_k) = Cx(t_k) + Du(t_k) \tag{2.17}$$

We do this by assuming the state of our system is given at some sampling time t_k, and the state at some future time t is given by solving 2.14.

$$x(t) = e^{A(t-t_k)}x(t_k) + \int_{t_k}^{t} e^{A(t-s')}Bu(s')ds' \tag{2.18}$$

which means that at the next sampling time we will get

$$x(t_{k+1}) = e^{A(t_{k+1}-t_k)}x(t_k) + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s')}Bu(s')ds' \tag{2.19}$$
$$= e^{A(t_{k+1}-t_k)}x(t_k) + \int_{t_k}^{t_{k+1}} e^{A(t_{k+1}-s')}ds'Bu(t_k) \tag{2.20}$$
$$= \Phi(t_{k+1}, t_k)x(t_k) + \Gamma(t_{k+1}, t_k)u(t_k) \tag{2.21}$$

This is true for a system that holds the value of the last received signal until a new signal has arrived, so called zero-order hold sampling. Here we have

$$\Phi(t_{k+1}, t_k) = e^{A(t_{k+1}-t_k)} \tag{2.22}$$
$$\Gamma(t_{k+1}, t_k) = \int_0^{t_{k+1}-t_k} e^{As'}ds'B \tag{2.23}$$

The matrix exponential is commonly defined by

$$e^{tA} = I + \frac{tA}{1!} + \frac{(tA)^2}{2!} + ... \tag{2.24}$$

where the terms of the series tend towards a nil value [3] .

The discrete state space model can thus be calculated from the continuous-time model that has been derived above. The representation of the system in continuous time was

$$\begin{bmatrix} \dot{p_1} \\ \ddot{p_1} \\ \dot{p_2} \\ \ddot{p_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\frac{k_s(p_1)}{m_1} & -\frac{d(p_1)}{m_1} & \frac{k_s(p_1)}{m_1} & \frac{d(p_1)}{m_1} \\ 0 & 0 & 0 & 1 \\ \frac{k_s(p_1)}{m_2} & \frac{d(p_1)}{m_2} & -\frac{k_s(p_1)}{m_2} & -\frac{d(p_1)}{m_2} \end{bmatrix} \begin{bmatrix} p_1 \\ \dot{p_1} \\ p_2 \\ \dot{p_2} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} F \tag{2.25}$$

This is put into the equations [2.22] with the understanding that $t_{k+1} - t_k$ is the same as the sample period h. Using the definition of a matrix exponential given by Eq. [2.24] the equations become

$$\Phi = e^{hA} = I + hA + \frac{(hA)^2}{2!} + ... \tag{2.26}$$

$$\Gamma = \int_0^h e^{sA} dsB = \int_0^h I + sA + \frac{(sA)^2}{2!} + ...dsB \tag{2.27}$$

These equations will not be solved by hand since the symbolical value of the matrices in general only lends itself to infinite evolutions, which is left as an exercise for the reader if so inclined. They will however be evaluated at a later point when numerical values are acquired, making them converge towards a constant value.

However, these equations are computationally very heavy due to integrals and the heavy matrix exponential. Therefore an approximation can be used, using the fact that for small timesteps, the equation becomes

$$e^{Ah} \approx I + Ah \tag{2.28}$$

which means that the approximative solution is

$$x_{k+1} = (I + Ah)x_k + hBu_k \tag{2.29}$$

Another approximation that can be used is the Tustin transform seen below

$$e^{Ah} \approx (I + \frac{1}{2}Ah)(I - \frac{1}{2}Ah)^{-1} \tag{2.30}$$

which preserves stability as well as instability of the discretized system from the continuous-time model. Once the matrices have been acquired two important tests have to be performed to check the feasibility of the project,

[3] if (tA) has its eigenvalues strictly in th left half plane.

which is to check if the system is *controllable* and *observable*. Controllability is loosely defined as "the ability to move a system around in its entire configuration space using only certain admissible manipulations", but can change depending on the setting. This can be translated to mean: is it possible for the system to reach all of the defined space points, with the defined input signals? For a discrete system, the degree of controllability is defined by the **controllability matrix [2]**.

$$W_c = [\Gamma, \Phi\Gamma, ..., \Phi^{n-2}\Gamma, \Phi^{n-1}\Gamma] \tag{2.31}$$

where n is the dimension of the Φ matrix. If the Gramian has full rank, i.e. $rank(W_c) = n$ then the system is *controllable*.

Similarly there is a test to perform to discern whether the system is *observable* or not. A system's internal state variable are not always measurable due to different reasons, such as the sensors not being implemented, or the state might be purely fictional. Thus, to be able to control around an unknown variable, one must make it known, by either measuring it or estimating it. A ubiquitous way of doing this is through observers, most commonly through a Kalman observer. This observer takes the known outputs and inputs and makes an estimation based on the proposed model of the system. However, due to how the system is modeled, this might not always be possible, since some states might be too far removed mathematically from either output or input to be estimated, i.e., some internal states may not affect the output directly, or too weakly to be estimated. Thus one must check the systems observability to see whether it is possible to implement an observer or not. This is done via the **observability matrix [2]**.

$$W_o = \begin{bmatrix} C \\ C\Phi \\ \vdots \\ C\Phi^{n-1} \end{bmatrix} \tag{2.32}$$

where once again, the observability of all the states hinges on whether $rank(W_o) = n$ where n is the dimension of the Φ matrix. Should this condition be fulfilled, an observer can be implemented in such a way that all states become known.

When moving from the continuous-time to the discrete-time domain it is important to note that the stability of the system has new criteria. When the stability region is transformed from the s-plane unto the z-plane it is moved from the left half plane into the unit circle, showed in Figure 2.3.

To understand this, one must think about the nature of a continuous system versus a discrete system. For a continuous-time system, feedback rendering at least one closed-loop system pole in the right-half plane (positive real part) creates a runaway (i.e., an unstable) system which, with any kind

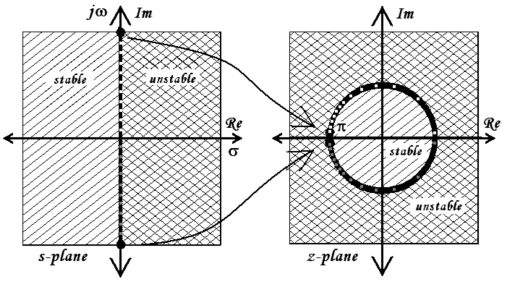


Figure 2.3 The stability region of the s-plane and the z-plane. [*https* : *//www.digikey.com/eewiki/display/LOGIC/IIR+Filter+Design+in+VHDL+Targeted+for+18−Bit%2C+48+KHz+Audio+Signal+Use*]

of positive feedback-loop tends towards infinity. Within a discrete system, this is the same if in every step, the increase in the system is larger than one, corresponding to at least one system pole outside of the unit circle. Modes with poles that in magnitude are smaller than one will eventually die off, since it diminishes with each consecutive step. This is true for the bilinear case, but other discretizations have different stability regions, such as Tustin with a circle in 0.5 with a radius of 0.25, and Euler with a plane left of 1 [**2**]. It is also important to note that each point on the left hand plane is not singularly mapped unto the circle, but many points might share the same point within the circle.

Another thing to note is that the zeros and poles of the discrete system is not solely dependent on the continuous system, but also on the sampling period h. An example from [**2**] illustrates this. Consider the continuous-time system

$$\frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + w_0^2} \tag{2.33}$$

where ω_0 is the natural frequency, ζ the dampening constant and s the Laplace variable. The poles of the corresponding discrete-time system are given by the characteristic equation

$$z^2 + a_1 z + a_2 = 0 \tag{2.34}$$

where

$$a_1 = -2e^{-\zeta w_0 h} cos(\sqrt{1-\zeta^2} w_0 h) \tag{2.35}$$

$$a_2 = e^{-2\zeta w_0 h} \tag{2.36}$$

Here it is clear that with different sampling period the poles are moved. Thus a suitable choice of sampling period can mean the difference between a stable

and unstable system. This usually comes down to choosing the sampling period to be within the range

$$\omega_0 h = 0.2 - 0.6 \tag{2.37}$$

where ω_0 is the desired natural frequency of the closed loop system. This implies that we are sampling up to 20 samples per period of the dominating mode of the closed loop system. It is of course also important to consider the Nyquist frequency so that all important signals of the system are observed in their original range and not aliased into lower frequency signals.

2.4 System identification

There is no model that can be described as the perfect model for a system. To choose a suitable model, the purpose of the model must be known. Depending on what scale you are looking, different laws apply to a system. For an atom, friction is not something calculated, but if enough atoms are collected and moving them is attempted, friction immediately makes itself known. Thus, choosing a proper model is tantamount to taking a first step towards success. However, a model might not be known beforehand, or *a priori*. This means that an identification of a system must be made, so as to try to discern what kind of mechanics it contains.

The topic of system identification is not a simple or straightforward one. Just as in choosing a model, choosing a proper way to try to identify your system might lead to success, or failure if the proper precautions are not taken when analyzing the data.

It is always helpful to have some knowledge about the system a priori, however, this is not always needed due to the strength of estimation methods. One of the most commonly used ones is the least-squares estimation [**1**]. As the name suggests, it uses a quadratic series to minimize the estimation of a series of parameters. The method bases itself on a posit that the system is constructed by a series of linear equations that describe the dynamics, namely

$$y = \sum_{1}^{N} \theta_n \phi_n(x) \tag{2.38}$$

where y is the vector of outputs, θ_n the unknown parameters to be estimated and $\phi_n(x)$ are known functions. The goal is to estimate the parameters in such a way that a reconstruction of the outputs $\hat{y}$ comes as close as possible to the actual output y. It is assumed that all measurements have the same precision, which by the principle of least squares, means that the parameters should be chosen so that the loss function

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{N} \epsilon_i^2 \tag{2.39}$$

where

$$\epsilon_i = y_i - \hat{y}_i \tag{2.40}$$

should be minimal. Using vector notation, this means that the loss function is

$$J(\theta) = \frac{1}{2} \epsilon^T \epsilon = \frac{1}{2} ||\epsilon||^2 \tag{2.41}$$

The equation

$$\hat{y} = \Phi\theta \tag{2.42}$$

has, with the goal to minimize $||\epsilon||^2$, the solution

$$\theta = \hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T \hat{y} \tag{2.43}$$

A common notation is

$$\Phi^\dagger = (\Phi^T \Phi)^{-1} \Phi^T \tag{2.44}$$

where $\Phi^\dagger$ is called the *pseudoinverse* of Φ.

This is a good way to identify systems that are linear and unknown in the parameters. However, it is also necessary to properly analyze the parameters post identification. Since this method is purely trying to estimate parameters linearly dependent on the input data, it can give false results. The variance and mean of the parameters are important to take note of, so as to see whether it might be a good estimation or not. If the variance is low, there is a higher probability that the parameters might be true. Should the variance be high, it might be proper to try to increase the order of the system so as to see if it might be a higher order system. However, if the order of the model is increased, the risk of *overfitting* increases. The more parameters you add the better the data might fit on the acquired data set, but quickly diverge on data outside the region.

Another way to analyze the estimated parameters is to feed the estimated system the input data that was used to estimate the system and see how the output differs from the true output data of the system. To see if the model has been overfit to the data set, a different data set should be used. A proper model will then still be able to closely approximate the true output, whereas an overfit model will be wildly divergent from the true value, which gives a good clue about the order of the model.

Another way of trying to identify a model is estimation using a grey-box model[4]. This is when some knowledge of the system is known a priori, e.g., the structure of the general mechanics of the system or some of the parameters are known. This means that there are only some parameters left to estimate of the system. The confidence in the model must then be high, since if the model structure is wrong, the estimation will be very off. It thus becomes very important to verify the estimated model using a few datasets.

The same is true for another method called black-box estimation[5]. This is when no prior knowledge of the system exists, and all parameters are free. The estimator tries to model a system so that the estimation fits the training data as closely as possible. Usually the only thing chosen is estimation method and order of the system.

Another valuable asset in system identification is the magnitude squared coherence estimate. Consider two signals $u(t)$ and $y(t)$. The measure of the

[4] `https://se.mathworks.com/help/ident/ref/greyest.html` Visited: 051218

[5] `https://se.mathworks.com/help/ident/ug/black-box-modeling.html` Visited: 051218

correlation between these signals, the cross spectra, i.e., how much of one is contained in the other signal, is defined by

$$\Phi_{yu}(w) = G_T(e^{iwT})\Phi_u(w) \tag{2.45}$$

where $\Phi_u(w)$ is the linear spectrum of the signal, i.e. how strong the signal is at different frequencies and G_T the transfer function from u to y. The definition for the linear coherence[6] is

$$C_{yu} = \frac{|\Phi_{yu}|^2}{\Phi_u \Phi_y} \tag{2.46}$$

This equation can only reach a maximum value of unity, indicating linear correlation, i.e. the signals are the same, and a minimum value of zero, indicating that they are linearly unrelated. This tool can be used to see how reactive a system is at different frequencies, and thus if the system has a good estimation at these frequencies. If there is no, or weak, correlation between input and output of a system, it poses a greater challenge for any estimator to get close to a true system. Hence the search can be closed off to the strongly correlated spectrum, or new tests devised so as to reach a higher correlation.

System identification today has been made a lot easier due to the introduction of powerful computational programs that have done away with the need for complex and tedious calculations to be done by hand. This has shifted the focus from mathematical operation to the understanding and philosophy of the topic. An excellent treatise on this is found in Glad&Ljung, Chapter 13 [**1**] and Johansson [**12**].

[6] https://www.fil.ion.ucl.ac.uk/∼wpenny/course/course.html Visited: 051218

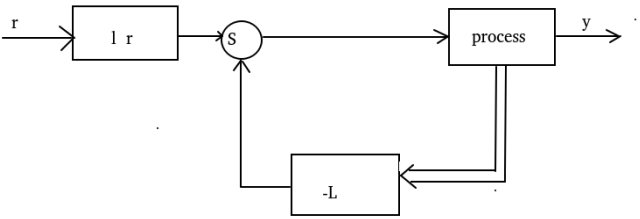


Figure 2.4 The control structure of state feedback control

2.5 State-feedback control

The most common way of controlling processes in industry is by using the simple P- or little more advanced PI or PID controller. They are easy to implement and often works sufficiently well, in particular when used in cascaded structures. However, when trying to implement control to behave in a specific way, with hard demands on control, it might not be the best choice.

A way to achieve a more general control structure is by using state feedback control [**4**]. This is based on the assumption that all pertinent states of the system are known, either by measuring them or using an observer to estimate. The regulator is created by giving the reference signal to the system some gain l_r, and putting the states of the system through the feedback vector L as can be seen in Figure 2.4.

The equations for the system then become

$$\dot{x} = Ax + B(l_r r - Lx) \tag{2.47}$$

$$y = Cx \tag{2.48}$$

which equates

$$\dot{x} = (A - BL)x + Bl_r r \tag{2.49}$$

This means that the dynamics of the closed-loop system then are decided by the choice of feedback vector L, thus the dynamics can be transformed to become as wished. For the system to be able to controlled this way, the system must be *controllable*, which is verified by the controllability gramian or controllability matrix as mentioned above. This ensures that all states of the system are controllable and don't reach a state from which they cannot be recovered.

2.6 Kalman filter

To be able to control a system efficiently, the states of the system must be known. This may be a problem when not all desired outputs of the system are known. The same can be said for the states of the system, especially when a state-feedback control is preferred. Thus the need for proper observers to be a part of the control structure.

One of the most famous, and ubiquitous, observers is the Kalman Filter.

It's based on the presumption that both the control signal u and the output signal y are available, as well as the dynamics and parameters of the system being known. It uses linear quadratic estimation (LQE) to estimate the state of the system, as well as give a prediction for the next update.

A discrete time system of the model has the dynamical equations as[7]

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y &= Cx(k) \end{aligned} \tag{2.50}$$

Using these equations, another term can be inserted, which is related to the Kalman Gain K, and the error between "true" and estimated output signal

$$\hat{x}(k+1) = A\hat{x} + Bu + K(y - \hat{y}) \tag{2.51}$$

$$\hat{y} = C\hat{x} \tag{2.52}$$

Combining these equations, the Kalman estimation of the state vector becomes

$$\hat{x}(k+1) = (A - KC)\hat{x} + Bu + Ky \tag{2.53}$$

This means that the evolution of the error is

$$\tilde{x}(k+1) = x(k+1) - \hat{x}(k+1) = Ax + Bu - A\hat{x} - Bu - KC(x - \hat{x}) = (A - KC)\tilde{x} \tag{2.54}$$

This means that the attributes of the filter can be decided by choosing the Kalman Gain K arbitrarily. However, the choice of the gain might be given more thought, so as to choose the gain optimally. In the discrete application of the Kalman filter there is an update algorithm where the initial covariance and the confidence in either model or measurement is chosen. In these equations the first part, Eq. 56 & 57, is the a priori prediction of the state estimate and error covariance $P_{k|k}$

$$\hat{x}_{k|k-1} = A\hat{x}_{k-1|k-1} + Bu_k \tag{2.55}$$

$$P_{k|k-1} = AP_{k-1|k-1}A^T + Q_k \tag{2.56}$$

[7] Note that in this chapter we use the system matrix representation (A, B, C) instead of (Φ, Γ, C) also for disctrete-time systems.

Here Q_k is the covariance of the process noise, and P_k the error covariance matrix. To obtain the posteriori estimations the following equations are used

$$\tilde{y}_k = y_k - C_k \hat{x}_{k|k-1} \tag{2.57}$$
$$S_k = R_k + C_k P_{k|k-1} C_k^T \tag{2.58}$$
$$K_k = P_{k|k-1} C_k^T S_k^{-1} \tag{2.59}$$
$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{2.60}$$
$$P_{k|k} = (I - K_k C_k) P_{k|k-1} (I - K_k C_k)^{-1} + K_k R_k K_k^T \tag{2.61}$$
$$\tilde{y}_{k|k} = y_k - C_k \hat{x}_{k|k} \tag{2.62}$$

Here y_k is the observation that is made, with C_k as the observation matrix, S_k is the innovation covariance and R_k the covariance of the measurement noise. These equations make use of *optimal* Kalman gain, K_k as calculated by iterative estimation and prediction by the filter.

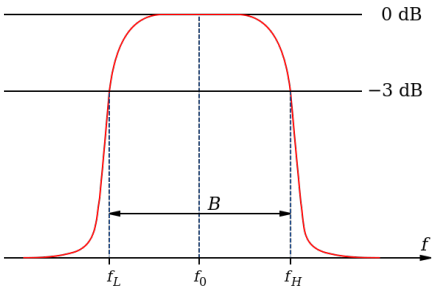


Figure 2.5 Band pass filter, where f_L and f_H are the frequencies at which the signal is attenuated to -3dB. B is the bandwidth of the filter, i.e. which frequencies are passed through. [$https://en.wikipedia.org/wiki/Band-pass_filter$]

2.7 Notch filtering

When analyzing a system, an integral part of it is the analysis of the frequency functions of the system. It can reveal a lot of behaviors about the system, such as dead zones, or regions in which the system becomes unstable. To avoid these zones, a proper control strategy must be implemented.

However, another way to achieve the same goal is to make sure the system gets no such signals in the first place. This can be done via filtering of the input signals to the system. A very common implementation is low-pass or high-pass filters in electrical circuits, to make sure a circuit behaves in the desired way.

In some systems, only some signals are "safe" to send through, which means that disturbances that can introduce frequency components outside these regions must be filtered away. A filter used to eliminate signals outside a desired range is called a *band-pass filter*, as can be seen in Figure 2.5. In this filter only signals in the frequency range f_L and f_H are permitted through the filter. However, what if there is a special frequency that is explicitly not desired? Then an inverse of this filter is needed. This is what is known as a *Notch filter*, or commonly, *Band-stop filter*. A visualization of how such a filter might look can be seen in Figure 2.6 Here the range between f_1 and f_2 is attenuated so as to not enter the system at all. The mathematical definition for this filter is formulated as

$$H(s) = \frac{s^2 + w_0^2}{s^2 + 2w_c s + w_0^2} \tag{2.63}$$

Here w_c denotes the width of the rejection band, i.e. the width of the range $f_1 - f_2$, and w_0 the frequency that is to be attenuated [**7**].

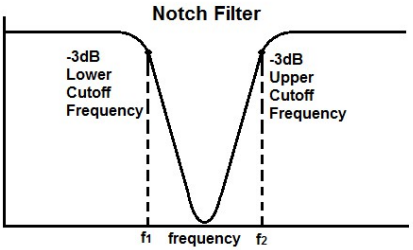


Figure 2.6 Band stop filter. [*https* : *//oscarliang.com/filter* − *betaflight/*]

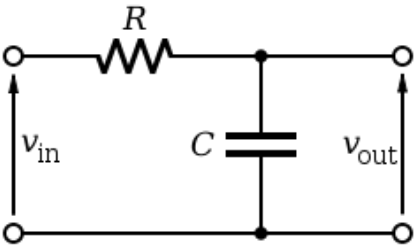


Figure 2.7 Low pass filter in electronics. [*https* : *//en.wikipedia.org/wiki/Low* − *pass*$_f$*ilter*]

2.8 Low-pass filter

In an ideal world the signal in or out of a system is always only the signal that is of interest with the properties that were intended. However, in a non-ideal world properties that are not desired arise, in the way of noise and e.g., load disturbances. Examples of this is atmospheric noise on the radio or in speakers. To get rid of this noise some kind of filter must be implemented.

There exists a plethora of different filters. However, the challenge comes in choosing the right one. For slow moving drifts and errors, a high-pass filter is often the proper choice, as only the high-frequency signals are passed through. Conversely, for high frequency errors a low-pass filter is the proper choice.

A straight-forward implementation of the low-pass filter is found in electronics, where high frequency signals might not be wanted due to its destructive properties to the hardware. It is realised by a resistance in series with the load, and a capacitor in parallel with the load, as seen in Figure 2.7.

From this circuit diagram, along with elementary electronics equation and some simplifications, the equation for the system becomes

$$v_{in}(t) - v_{out}(t) = RC\frac{dv_{out}}{dt} \tag{2.64}$$

substituting v_{in} with the sequence $x_1, x_2, ...x_n$ and v_{out} with $y_1, y_2, ..., y_n$ and moving to discrete time, the equation becomes

$$x_i - y_i = RC\frac{y_i - y_{i-1}}{\Delta t} \tag{2.65}$$

and rearranging this gives

$$y_i = \alpha x_i + (1-\alpha)y_{i-1} \text{ , where } \alpha = \frac{\Delta t}{RC + \Delta t} \tag{2.66}$$

the properties of this filter are the time constant RC and the sampling time Δt. The cutoff frequency of this filter is defined by

$$f_c = \frac{1}{2\pi RC} \text{ which means } RC = \frac{1}{2\pi f_c} \tag{2.67}$$

This filter has the magnitude of unity up until the cutoff frequency where the magnitude is -3dB. Thus this filter is a good choice for when the frequency of the signal into the system is known and all noise entering is of high frequency.

3

Method

The entire process of identification of the system and validation can be condensed into the list below

1. Run tests
2. Acquire Data
3. Remove frictional component from input torque
4. Filter the signals through a low-pass filter
5. Estimate the system:
    a) using grey-box estimation
    b) using black-box estimation
6. Discretize system
7. Validate system by running the same input through the estimated system, as well as validate using other data

This will show whether the estimated system is close to reality or not. The estimated system will then be used in a Kalman-filter so as to be able to control the process in a sufficient manner.

3.1 Experiments

Experimental setup in Switzerland

The basis for the experiments and control design is the process created by Güdel known as the 2-axis ZP-4 Gantry. [1]

[1] `http://www.gudel.com/products/linearaxis/zp`

Figure 3.1 The Swiss Process - a 2-axis ZP-4 gantry robot. .

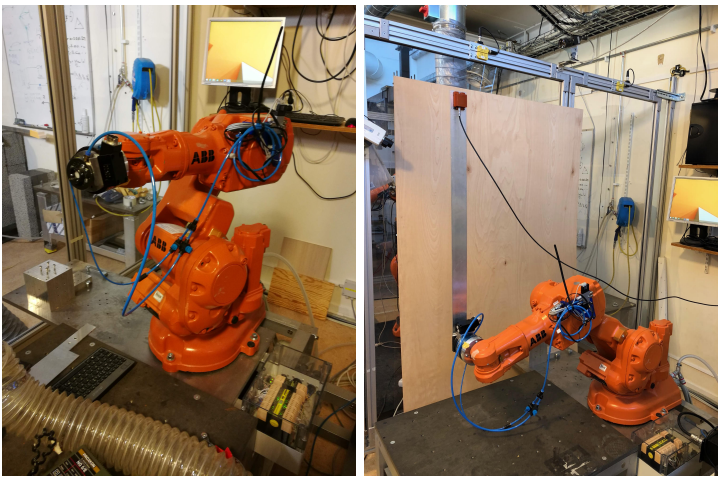
This process moves along two axes. The y-axis which is the position of the here-after called "cart" on the beam. The z-axis which is the height position of the here-after called "tool". The x-axis is the axis which goes "through" the picture. The y- and z-axes are controlled by two individual electrical motors, which means that the position of cart and the height of the tool can be controlled independently. This process is controlled via the Beckhoff TwinCat3 software which runs on a computer, which communicates with an IndraDRIVE that sends the control signals to the motors, and receives signals from the different sensors of the process.

The motors controlling the process are produced by Rexroth, with the motor for the y-axis being the MS2N07 and the z-axis the MSK.

This robot has a few different sensors available to use. For the cart on top there is the position, velocity and acceleration, derived from the encoders on the motor, and converted to absolute position and speed in millimeter as well as acceleration. To measure the lower mass a laser positioning device by the brand Baumer, model OADM 12i6460/s35a, was used.

This measured the position from the laser measuring device to the lower mass CMP2. These measurements were available in real time and synchronised with the gantry sensors through the TwinCat software. In TwinCat we had the possibility to record these measurements for later evaluation, as well as use them in controls. The TwinCat software was connected to a drive for the system, an IndraDRIVE. This drive had the option to send references to position, velocity or straight to the torque. Within the drive a cascaded control loop is implemented so as to follow these references.

If the drive was set to receive references to position, it generates reference trajectories for the robot to follow, so as to have a smooth movement.



(a) The Swedish process.

(b) The Swedish process with a flexible beam attached. An IMU/accelerometer is mounted at the end of the beam.

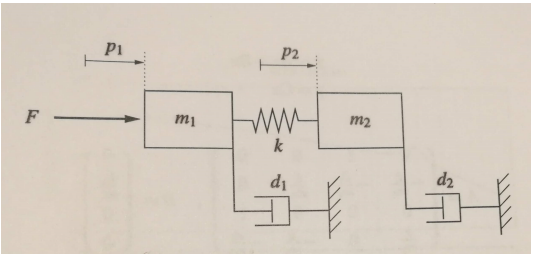
Figure 3.2 Figures of the Swedish robot setup

Experimental setup in Sweden

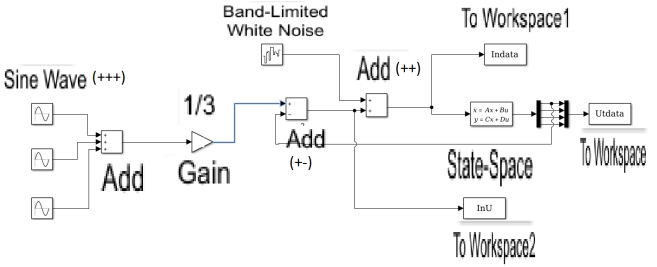
For some of the identification another experimental setup was needed, and such a robot was equipped in the Robotics Lab at LTH, as seen in Figure 3.1.

This robot was locked in all joints, with the top-most arm placed parallel to the ground. The tool-gripper was equipped with a long metal piece of some unknown elasticity, with a weight placed at the outer-most tip. The aim of this was to simulate the system in Switzerland, by rotating the robot horizontally and keeping all other joints locked. This excites vibrations in the outermost weight, in similar fashion to the ZP-4. During tests, data could be collected from the robot's sensors and actuators. However, a major difference is the lack of a distance sensor on the added weight, marked by the red rectangle in Figure 3.1. This was due to no distance sensor being available to meet the standards set by the tests. Instead, an accelerator was placed at the edge of the arm and the data from it analyzed instead.

The friction was also analyzed by commanding a triangle wave as torque reference to identify breakaway torque and the torque at which the system came to rest again. A series of constant speed tests were also run to try to identify the frictional torque curve.



(a) The ideal known system [1]



(b) Simulink diagram for the ideal system

Figure 3.3 Figures of the ideal system and its simulation setup

Simulation setup of idealized system

To gather some additional data on the estimation methods a system with all parameters and dynamics known was used. This system can be seen in Figure 3.3.

For the different tests the feedback and/or the noise component was added or removed, so as to see how the signals affected the output of the different estimations and analysis methods. The input is a sum of three different sinusoids, with different frequencies, combined and gain-adjusted so as to alter between maximum unity and minimum negative unity.

Friction

To estimate the friction a few different positions, as seen in Figure 3.4, were tested. During these tests, CMP2 was raised until it was at top position, i.e. closest to the cart, CMP1.

For these positions, the cart was run at a constant velocity, so as to see whether the position had any influence on the frictional component. To further collect data regarding the frictional component, a series of different speed tests were ran, so as to be able to discern the torque-velocity relation.

All these tests were ran at a constant speed to see what the torque necessary to neutralize friction was.

After that, the torque to force constant had to be tested, which meant a series of increasing speed tests, still with CMP2 raised.

Full system

After this, CMP2 was lowered to full extension in the set-up, i.e., 1200 mm extension of the beam carrying CMP2. The laser sensor from Baumer was set up to point at the middle of CMP2, to measure the distance to it. This sensor had an operating range of 16 to 120 mm, which meant that the ability to measure distance was limited to deviations within this range. However, the maximum resolution was somewhere around 0.12 mm, which meant that the accuracy was sufficient for the types of movement that were of interest.

The IndraDrive was first set to accept torque references. However, after testing this was shown to not be sufficiently tuned i.e. the loop did not reach the reference value in required time. This meant that the drive had to be set to point-references, which had sufficient tuning so as to follow the reference trajectory properly.

The signal sent to the system to excite the vibrations that were of interest was a sinusoidal signal with an increasing frequency, a so called *chirp* signal. This was used to provide input excitation of almost a continuum of frequencies in a certain frequency range.

Again, the tests were ran at the different positions seen in Figure 3.4. This was due to an assumption that the bending of the bearing beam due to the total mass of cart, tool and vertical beam being 473 kg being moved along the bearing beam. This was assumed to influence the resonant frequency of the system, due to the bending of the beam, which is correlated to the placement of the supports. Thus, tests had to be conducted so as to test this hypothesis.

The chirp-signal tests were run in such a manner that they were stopped when the torque reached maximum output value, to not damage the robot. This of course meant some limitation in the range in which the tests could be conducted. Another important issue was that the vibrations of CMP2 had to be limited in such a way that they did not leave the working range of the laser measurement device. This meant that the chirp signal used to excite these vibrations had to be limited in its amplitude to not leave this range.

F_c	F_s	F_v	v_s	δ_v	δ_s
0.606	1.43	2.21	0.00006224	0.4994	0.4962

Table 3.1 table

Values of frictional equation

3.2 Identification of friction and torque-friction relation

The friction has been assumed to be somewhat linear, however, that must be tested and verified to be true. This is done via torque identification.

The dynamical equation of a motor is

$$\tau_m = J\ddot{\theta} + \tau_f + \tau_{load} \tag{3.1}$$

where τ_{load} is assumed to be constant, τ_f the friction and $\ddot{\theta}$ the acceleration of the motor angle. From this we can see that when the velocity of the motor is constant then the remaining torque is purely used to overcome friction. Thus steady state velocity tests were run to identify the friction torque at different velocities and positions, and thus identify the dependencies. The tests revealed a non-linear dependency on velocity, one that could be identified to be a combination of the Coulomb friction, stiction, viscous friction as well as a Stribeck term. The resulting equation that was shown to fit the experimental data to 99% was

$$F_{fric} = F_c + (F_c - F_s)e^{-|\frac{v}{v_s}|^{\delta_s}} + F_v|v|^{\delta_v} \tag{3.2}$$

with identified parameter values as in the Table 3.1

The first step of identification was to run a series of single speed tests. These were run at the different positions indicated in Figure 3.4. This was done so as to test if there was any dependency on the position of the cart on the beam with regards to friction. However, these tests showed the same frictional torque at the same speed on the positions, which meant that the friction was not positionally dependent.

Thus a series of different constant speed tests were made. The speeds were chosen in small increments close to zero, as the friction varies a lot in the initial region, with larger steps being taken at higher velocities. The torque value for these speeds can be found in the Figures 3.6 & 3.5.

This data was used in conjunction with the previous model for friction in the Curve Fitting Toolbox in Matlab, which fitted the parameters of the equation so as to fit the data. The result of this is shown in 3.7 and in Table 3.1.

Thus we have an equation for the friction which seems to fit reality. Since the friction is now known we can start identifying the input parameter for the system. Since the torque is converted into force via the motor, we must

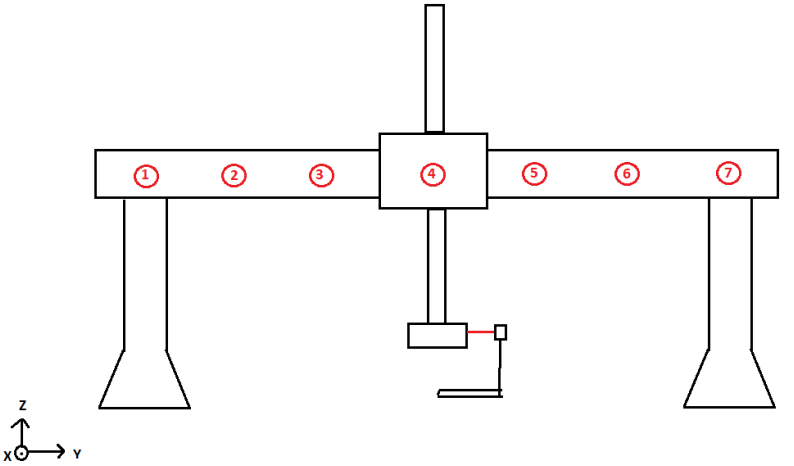


Figure 3.4 Model of testing setup, the numbers correspond to the different test positions

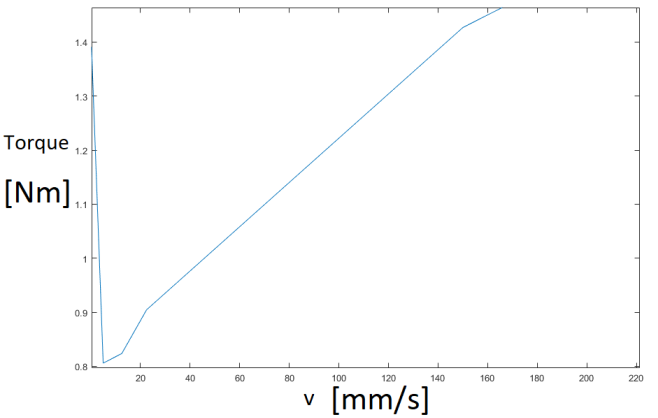


Figure 3.5 Zoomed in plot of the friction. [Nm] on y-scale, [mm/s] on x-scale

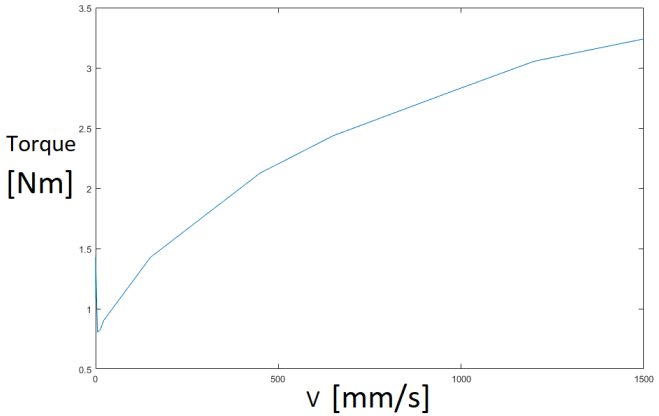


Figure 3.6 Plot of measured friction. [Nm] on y-scale, [mm/s] on x-scale

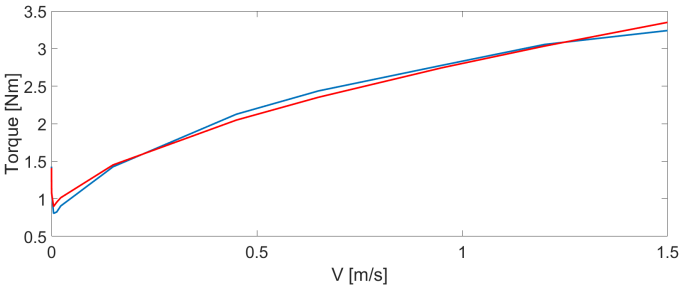


Figure 3.7 Measured friction plot in blue, versus fitted plot in orange. [Nm] on y-scale, [m/s] on x-scale

know the conversion parameter. A series of point-to-point motions were thus enacted, and the data from those recorded. Using the input torque with the friction subtracted we get the "true" input to the system. Thus we have the acceleration of the system and the input of the system. We can then use least-squares estimation to get an estimate of the conversion factor from torque to force.

The equation of the motion, with the lower weight CMP2 once again raised, becomes

$$M\ddot{p} = c(\tau_m - \tau_{fr}) \tag{3.3}$$

Where M is the total mass of the cart, vertical beam and CMP2.

The estimate is then achieved by letting $\Phi = \tau_{\mathbf{m}} - \tau_{\mathbf{fr}}$ and $y = M\ddot{\mathbf{p}}$. Thus

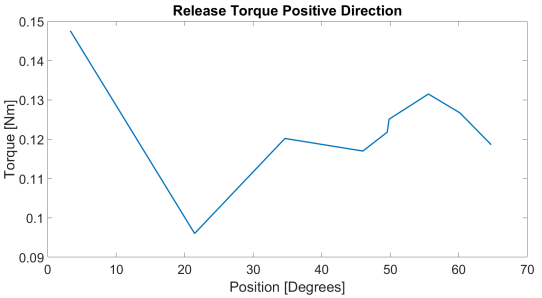


Figure 3.8 The torque needed to breakaway at different angles for joint 1 of the IRB140-robot. Positive direction.

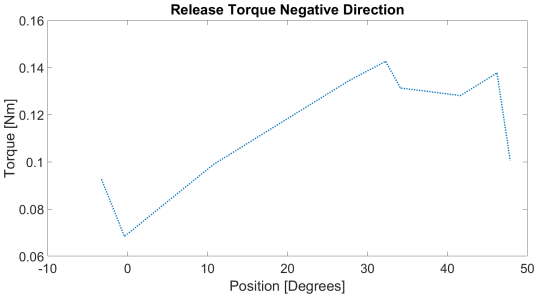


Figure 3.9 The torque needed to breakaway at different angles for joint 1 of the IRB140-robot. Negative direction.

the equation is, using the pseudoinverse,

$$(\Phi^T\Phi)^{-1}\Phi^T\mathbf{y} = c \tag{3.4}$$

Which yielded a value of $c = 1.7$. We now have the equations leading in to the system and can now start to test the system itself.

Fricion in IRB140

To try to analyze the breakaway torque for the Swedish robot a triangle wave was sent as a reference to the torque. The time at which the robot started to move was deemed as the point where the torque had overwon the stiction. The torque was analyzed both in positive and negative direction. The following graphs in Figures 3.8 & 3.9 have been flipped so as to ease understanding.

Clearly, going in the positive direction requires a higher input torque than going in the negative direction. Similarly when analyzing the force at which the robot comes to a rest is also quite different.

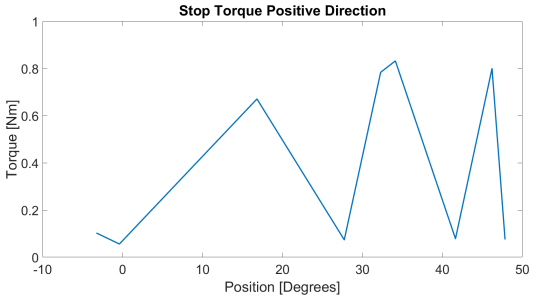


Figure 3.10 The torque at which the robot comes to a rest at different angles for joint 1 of the IRB140-robot. Positive direction.

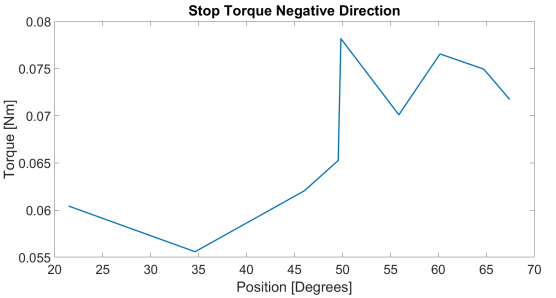


Figure 3.11 The torque at which the robot comes to a rest at different angles for joint 1 of the IRB140-robot. Negative direction.

After these tests were run, a series of constant speed tests were conducted. These were run in closed loop to try to reach a static speed. The mean of the torque at the static speed was measured to reach an average value of the torque at the different speeds. The result of these speeds is shown in Figure 3.12.

These tests were run with an incrementally diminishing speed and alternating directions. The tests that can be seen in Figure 3.12 had a midpoint at around 30 degrees. This curve changes drastically when another tests was taken into account that was run with a midpoint at 12 degrees [**11**]. This cumulative result can be seen in Figure 3.13.

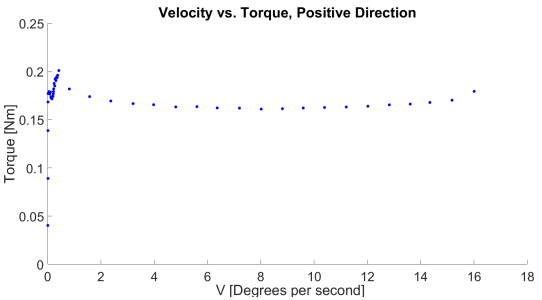


Figure 3.12 The frictional torque plotted against speed in positive direction for the IRB140-robot.

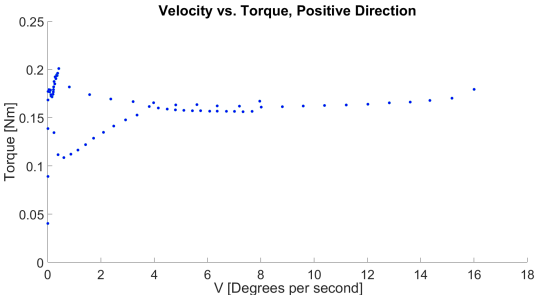


Figure 3.13 The frictional torque plotted against speed in positive direction for the IRB140-robot.

3.3 Analysis of full system identification

To be able to identify the full system, measurements of CMP2 were conducted.

This was done via the laser positioning tool from Baumer. This has a limited working range which meant that only small movements were possible.

The movements of the cart were done in a sinusoidal wave with 2 mm distance from endpoint to endpoint. These movements were at first slow and then gradually increased frequency to the point where amplified oscillations started occurring in the lower weight.

These tests were also run in the seven different positions, seen in Figure 3.4, so as to see whether there existed any positional dependency on the motion of the lower mass. The reasoning behind this suspicion is that the mass is not small, which means flexion of the bearing beam might occur, and thus impact the movement of the lower mass.

Once these tests had been conducted, the outcome was imported to the

System Identification Toolbox in Matlab, command *ident*. Here the data from the experiments was entered, once with torque as input and once with CMP1's position as input, and both with CMP2's position as output. Just using the raw data did not yield any valuable insight.

Pre-processing the data by filtering out high-frequency noise and removing means yields better working data. This data is after that used to estimate state-space systems to analyze how the system behaves. For the torque to position data, the average means were removed, thereafter band-pass filtered to allow signals in the range 0 to ∼66 rad/s. Thereafter, all positions, except position 4, were estimated to discrete state-space systems, using position 4 as validation data. This yielded a fit between estimation and validation data of 100%. The same process was applied to the data where CMP1 was used as input. The fit between estimation and validation data was again, 100%. The resulting frequency functions can be seen in Figures 3.3 & 3.3

It shows a clear static resonant frequency (at around 6Hz), for input CMP1 position and output CMP2 position. However, torque to CMP2 shows no such relation.

An attempt was made to use the Systems Identification Toolbox to find a state space of the total system, however, this could not be found, since there was no way to use all the data that was collected at the same time.

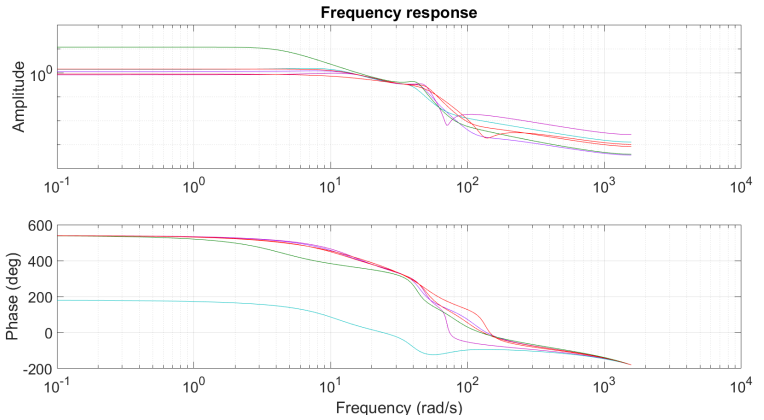
Thus a different approach was used, by trying to estimate via a grey-box model, and using the *greyest* command in Matlab.

To begin this estimation, tests had to be run on the discretization approximation method proposed earlier in Equation 24. These tests were run on a similar system, with the variables being set to fixed values, so as to be able to test it. The various different discretizations that were compared were zero-order-hold, Tustin, the approximation along with the plot for the system so as to compare them. The results can be seen below in Figure 3.15.

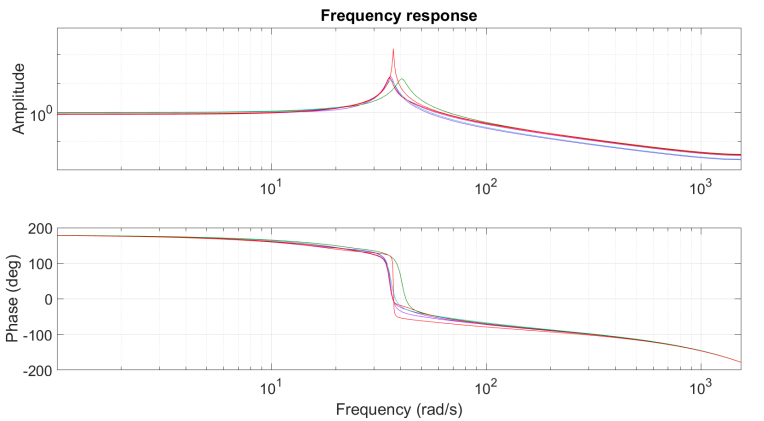
As can be seen, the plots of the various methods all lie along the same plot and can thus be seen as equivalent. Thus the next step is to test whether the grey-box estimation method in greyest can be used to properly estimate the system. This is again done on a known system, but with a few parameters set to other values, so that the system might be approximated. Data are generated via a Simulink model with a mixed sinusoidal input to the system, and at the beginning, without any noise. The results of estimation in discrete time and continuous time can be seen below in Figure 3.16.

This clearly shows a discrepancy in both estimations, however, the continuous-time model is clearly the better choice.

The next step is to introduce noise into the system to decide whether the estimator can handle the extra uncertainty. A band-limited white noise generator was introduced, with a limited power. The input signal before and after added noise can be seen in Figure 3.17 and the corresponding Bode diagram to the estimated system can be seen in Figure 3.18.



(a) Frequency function of estimated system with Torque as input



(b) Frequency function of estimated with CMP1's position as input

Figure 3.14 Plots of frequency function of estimated system, with rad/s as x-scale

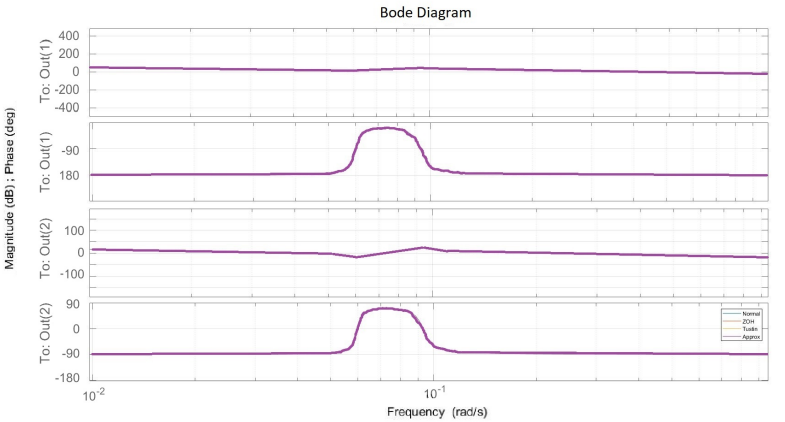


Figure 3.15 Bode plots of the different discretization methods. The standard method in *c2d*, zero-order-hold, Tustin and the approximation used in Equation 24.

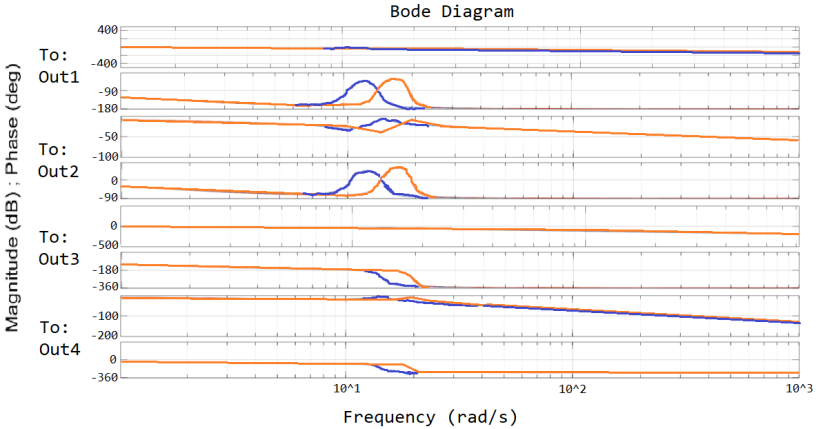
Obviously this means that the estimated system will perform pretty poorly initially as well as miss the resonant frequency with a good amount. To try to optimize the performance of the estimation, a low-pass filter will be added to the in- and outputs of the system. The insignal with noise, filtered and original signal can be seen in Figure 3.19 and the corresponding Bode-diagram to the estimated system can be seen in Figure 3.20.

This is a clear improvement in performance, however, still with some deviations. However, with the knowledge from the frequency plots in the System Identification Toolbox the validity of the estimated system can still be gaged.

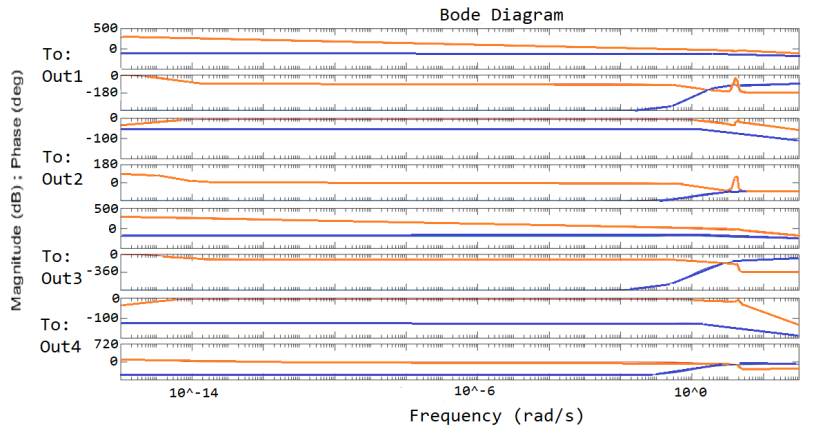
Another method to analyze the system is to use the magnitude squared coherence estimate, seen in Eq. 46, Matlab command *mscohere*. This measures how much of the input signal that the output signal contains, i.e. how much does the output correlate to what is put in to the system. A strong value, above 0.6, means that the system can be estimated well for the input and output signals that have been collected. Following figures are the log-scale plot of the coherence power (between 0 and 1), as a function of the frequency given by radians per second.

Since the input signals are of low frequency, i.e., less than 10 rad/s, the correlation is strong for those frequencies, but drops drastically after. The noise causes high tops on higher frequencies which are filtered away, as can be seen in Figure 3.21.

In Figures 3.22 the feedback is added, to create a loop of out the system, to see how this affects the coherence of the signals. As can be seen, the top



(a) Bode plot of estimation of continuous system.



(b) Bode plot of estimation of discrete system.

Figure 3.16 Bode plots of estimated systems, continuous and discrete.

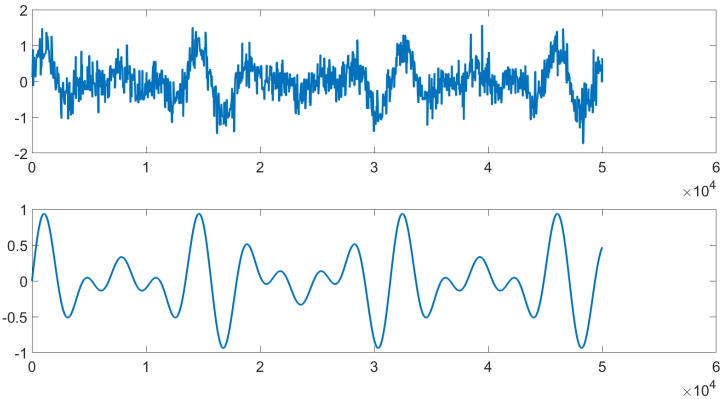


Figure 3.17 In-signal to the ideal system, with and without noise. Dimensionless, time-step on x-scale.

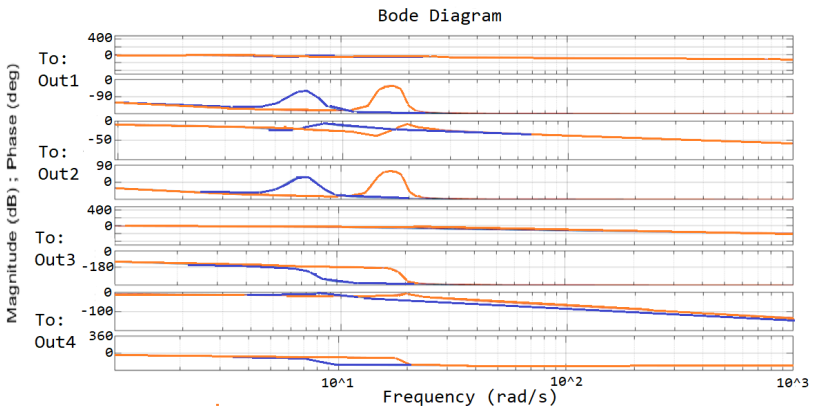


Figure 3.18 Bode plots of the estimated system with noise in input.

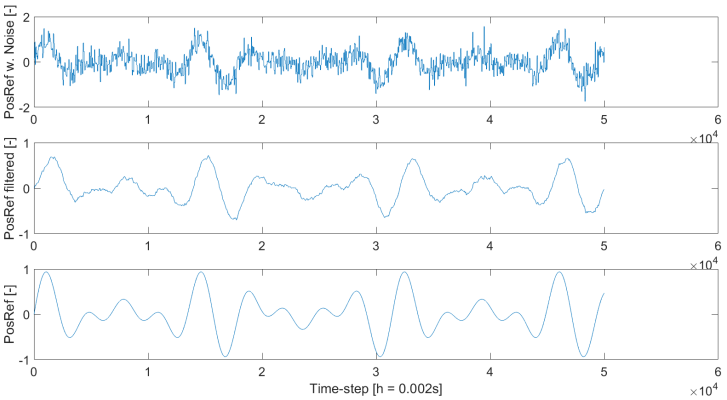


Figure 3.19 In-signal to the ideal system, with and without noise, as well as filtered signal. Dimensionless, time-step on x-scale.

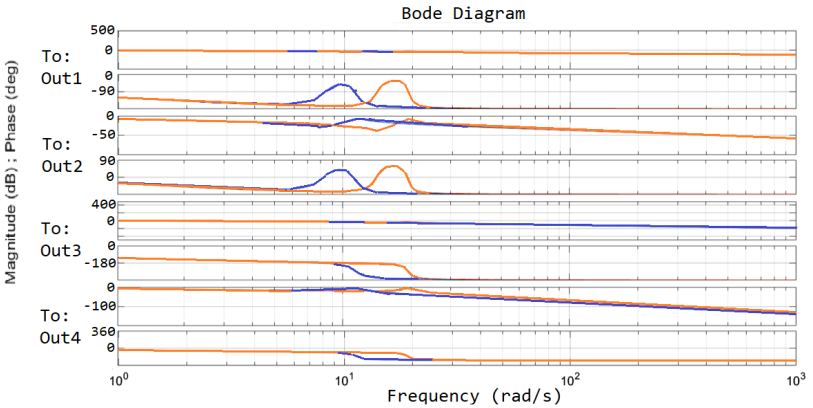
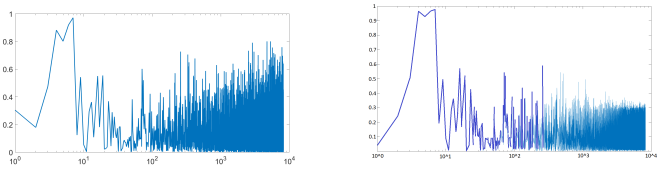
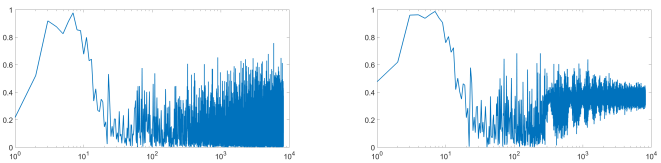


Figure 3.20 Bode plots of the estimated system with filtered signals



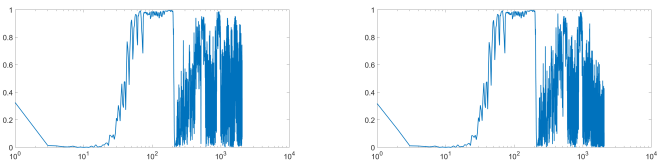
(a) With noise. (b) Low-pass filtered signal.

Figure 3.21 Log-scale coherence plots of ideal system in open loop. [rad/s] on x-scale.



(a) With noise. (b) Low-pass filtered signal.

Figure 3.22 Log-scale coherence plots of ideal system in closed loop. [rad/s] on x-scale.

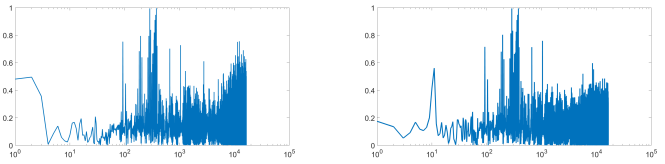


(a) With noise. (b) Low-pass filtered signal.

Figure 3.23 Log-scale coherence plots of Swedish system in open loop. [rad/s] on x-scale.

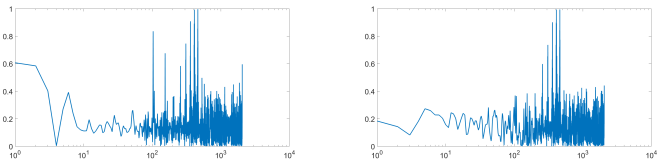
in the low frequencies is expanded, but the tops in high frequencies persists. When the signal is low-pass filtered however, another clear difference can be seen. There is now a raised line at higher frequencies. This can be attributed to the feedback, since the system sends high-frequency signals back into itself. To move on to the real system in Sweden, the plots can be seen below.

Since the system here was measured via accelerometer the top is moved further to the right. Slow movements cause almost no deviations in accelerometer data, hence the low value for the lower frequencies. The system input was also of a higher frequency than that of the ideal system. Had a distance



(a) Low-pass filtered Torque. (b) Frictional component removed.

Figure 3.24 Log-scale coherence plots of Swiss system in closed loop, low-pass filtered. [rad/s] on x-scale.



(a) Low-pass filtered Torque. (b) Frictional component removed.

Figure 3.25 Log-scale coherence plots of down-sampled Swiss system in closed loop, low-pass filtered. [rad/s] on x-scale.

sensor been used, a clear coherence could've been seen. It can also be seen that the higher frequencies do persist in high-value. This can be due to the noise in the accelerometer that is not filtered.

Finally, the Swiss system can be seen in the Figures 3.24 & 3.25 below.

This is with original sampling time, the down-sampled system (a tenth of the frequency), can be seen in the figures below.

Full system identification of IRB140

To analyze the dampening and spring constant of the arm on the Swedish robot the arm was bent to a 25mm distance from the middle and then let go. The accelerometer readings were collected and can be seen in Figure 3.26.

From this the decay as well as the period of the oscillation could be discerned. With this the poles of the system could be calculated, leading to poles at $0.2554 \pm 27.1412i$. A zoomed in picture of the beginning of the oscillations can be found in Figure 3.27.

Here clear overtones can be seen which indicate the existence of another pole pair.

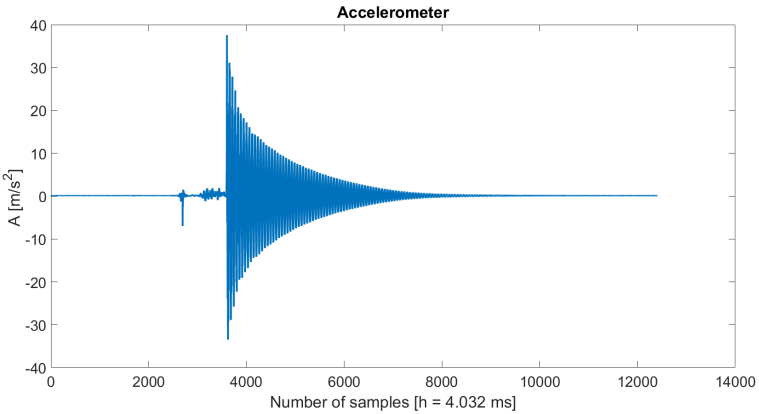


Figure 3.26 Accelerometer values.

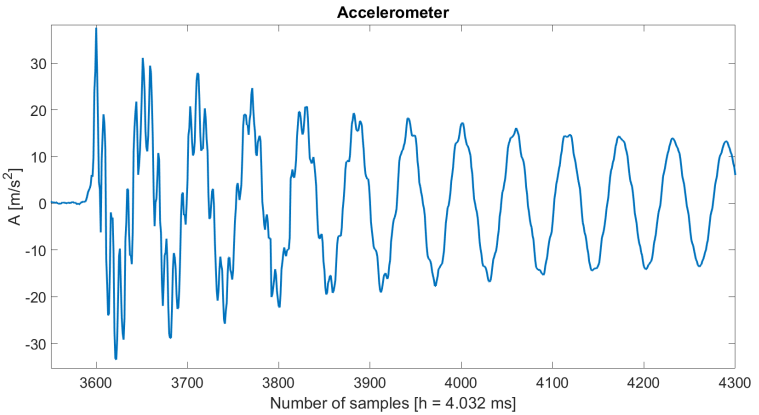


Figure 3.27 Accelerometer values, zoomed in.

3.4 Kalman Filter

The identified system was then used to set up a Kalman filter for the system, with only the "actual" input torque, the velocity and position of the upper mass as inputs to the filter. The initial idea was to implement this filter using a Simulink module in TwinCat. The reasoning for this was that a C++ module can be more efficient in real time settings with matrix calculations. Since the Kalman filter uses a lot of matrix calculations it is pretty computationally heavy. The implementation is done via an app for Matlab programmed by Beckhoff, where you can export a Simulink model to a .tlc extension, which can then be imported to TwinCat. However, time constraints and the difficulty in learning this method in a short amount of time turned us away from this option. Another method that was tested was a python to C++ converter. However, that also was proven to be a too high difficulty step to take.

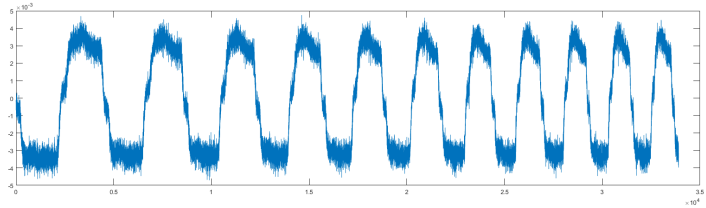
Finally, an app[2] for Matlab was found where you can export Simulink blocks directly to PLC (*Programmable Logic Controller*) code. A filter was thus programmed in Simulink and then exported via this app to Structured Text. This app optimizes the code so as to be a series of for-loops with vectors that makes the calculations light in regards to computations. However, this optimization also obfuscates the operations happening within the code, and it is not always clear how things happen.

The filter was initially programmed to have a dynamic Kalman gain matrix to achieve optimal gain as soon as possible. The filter was then run online with the equations so as to see whether it could be considered a good approximation of the system. This filter approximated the system very poorly, even after several attempts to change the noise and initialization matrices, as can be seen in Figures 3.28. It does not converge to the measure position, but stays at origo and deviates from there, as well as the estimations being too noisy and too large.

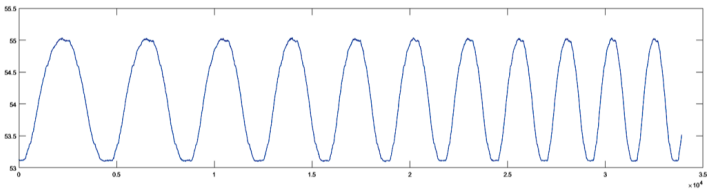
Hence, another method was needed. A static Kalman gain was then designed by pole placement. This pole placement did however prove itself to be difficult due to the values of the dynamical matrix. I strived to place the poles so that the bandwidth was sufficiently large so as to see the frequencies at which the system started to resonate. MatLab could however not place my poles exactly where I wanted them, and I was told that they may be as much as 10% off of their desired value. The MatLab command *place* was used to place the poles. Testing showed promising results initially, however, they did deviate at the higher frequencies, as seen in Figures 3.29 & 3.30.

This was as far as I came in Switzerland before the time was up. Obvi-

[2] `https://se.mathworks.com/help/mpc/ug/simulation-and-structured-text-generation-using-plc-coder.html` Visited: 051218



(a) Plot of the estimated position.



(b) Plot of measured position.

Figure 3.28 Plots of measured and estimated position of CMP2. y in [mm], time-step [h=0.002s] on x-scale.

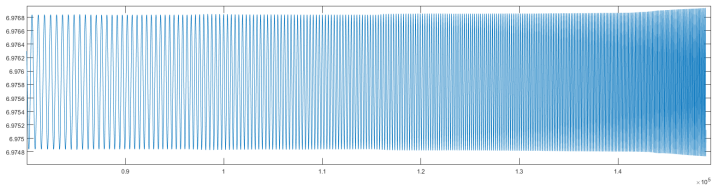


Figure 3.29 Plot of the estimated position of CMP2, distance on y-axis in [mm], time-step [h=0.002s] on x-scale.

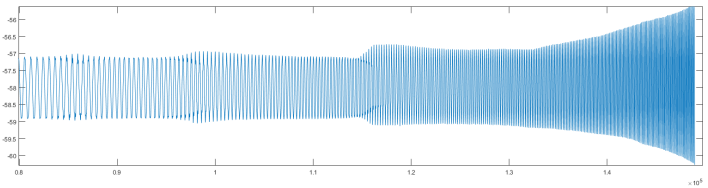


Figure 3.30 Plot of measured position of CMP2, distance on y-axis in [mm], time-step [h=0.002s] on x-scale.

ously, a better implementation of the filter was needed. However, if this is due to the model being too far off from reality or if the estimations introduced errors which have propagated through the process is something that needs to be analyzed. The pole placement was also non-ideal due to the failure of MatLab to place them in the desired locations. So there are aspects of the process that need to be analyzed and redone to reach a definite conclusion.

4

Results

The signals of the system that were used for the identification typically looked like Figure 4.1.

When used to identify the system, they yielded, for a continuous estimation that was discretized, the Bode plots seen in Figures 4.2 & 4.3. These estimations were done for tests conducted at position 4 & 7 (as seen in Figure 3.4).

When the approximated systems were simulated with the same input torque as the real system, and setting CMP1:s position & velocity to the real system at each revolution, they yielded the following graphs

Removing means, trends and filtering the signals from 0 to 100 rad/s in System Identification Toolbox, with position 4 as estimation data and position 7 as validation data, yielded the fit vector

-92.89%	95.55%	3.006%	95.33%

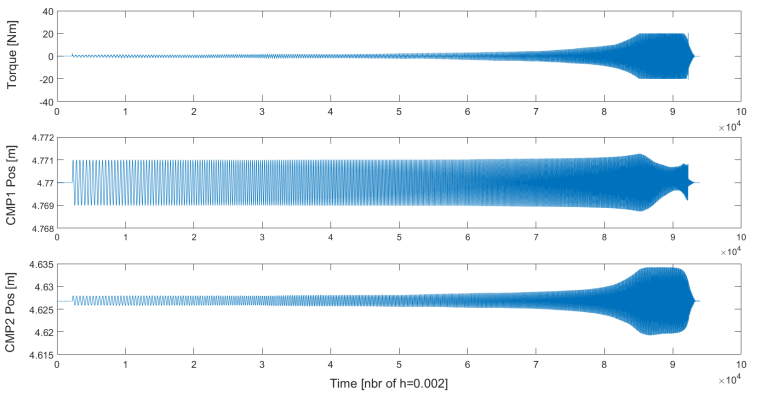


Figure 4.1 Input torque to the system in [Nm], along with position of CMP1 & CMP2 in [m]. All x-scales in time-step [h = 0.002s].

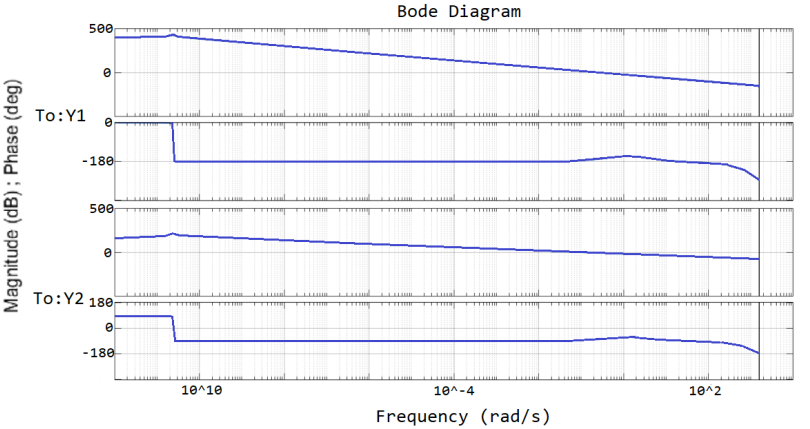


Figure 4.2 Bode diagrams of the estimated system with filtered signals for position 4 (Figure 3.4)

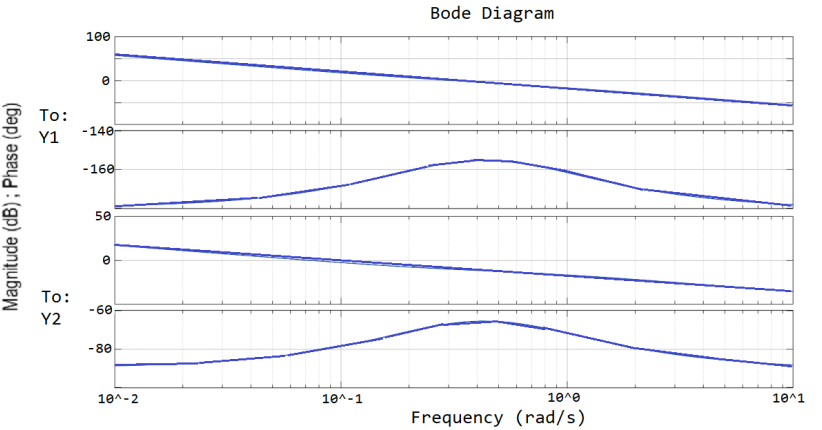


Figure 4.3 Bode diagrams of the estimated system with filtered signals for position 7 (Figure 3.4)

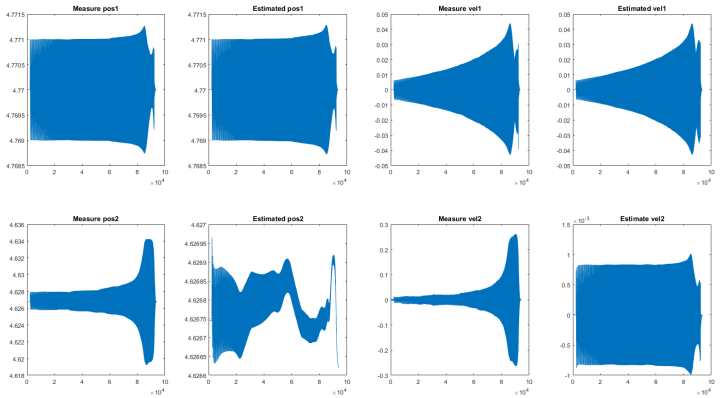


Figure 4.4 Simulations of the system, shown with the correct values to the left, and the simulated value to the right. Position 4 (Figure 3.4).

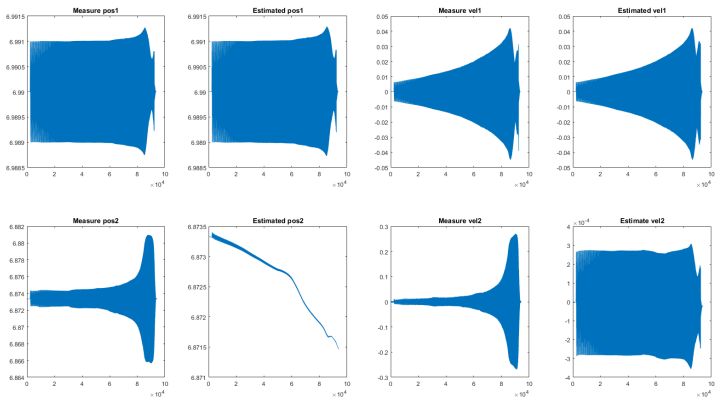


Figure 4.5 Simulations of the system, shown with the correct values to the left, and the simulated value to the right. Position 7 (Figure 3.4).

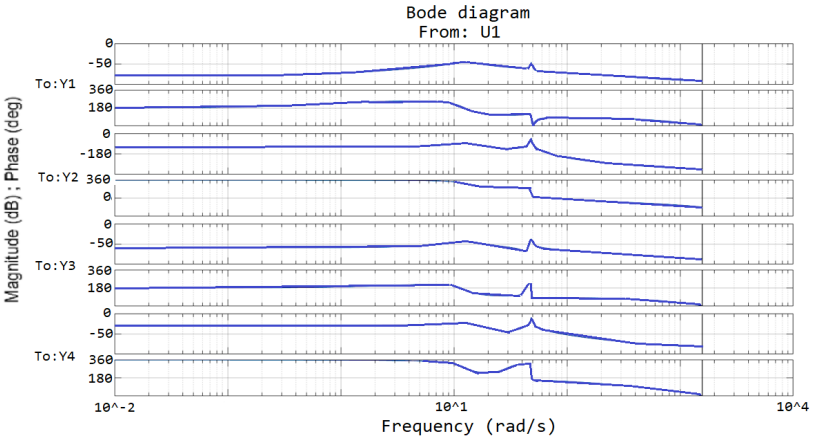


Figure 4.6 Bode-plot of the black box estimation

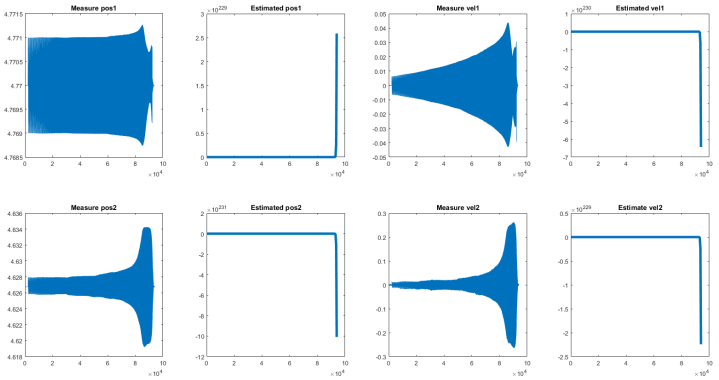


Figure 4.7 Simulation of the system with position 4 as data.

This black-box estimation in discrete time yielded the bode-plot seen in Figure 4.6

with the simulation yielding Figure 4.7.

The results for the down-sampled system to a tenth of the original sampling time, i.e., 50 Hz, yielded the results below in Figure 4.8 for a black-box estimation with the same inputs.

Finally, another grey-box estimation was done for the downsampled system which can be seen in Figure 4.9.

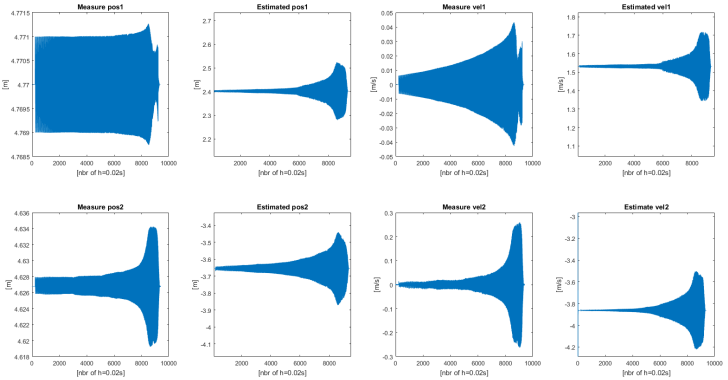


Figure 4.8 Simulation of the down-sampled black-box estimated system with position 4 as data.

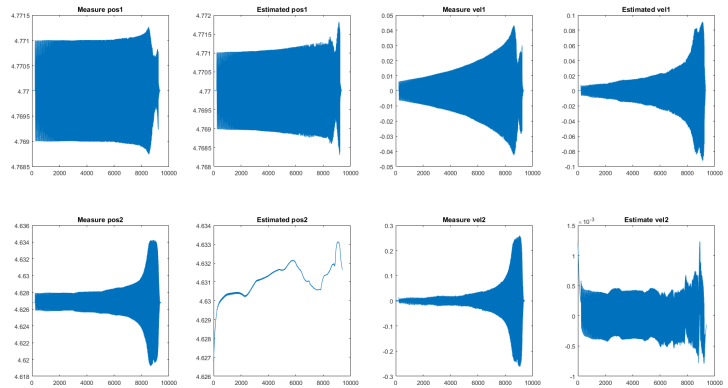


Figure 4.9 Simulation of the down-sampled grey-box estimated system with position 4 as data. Positions in [m], velocities in [m/s], x-scale in [nbr of h=0.02s].

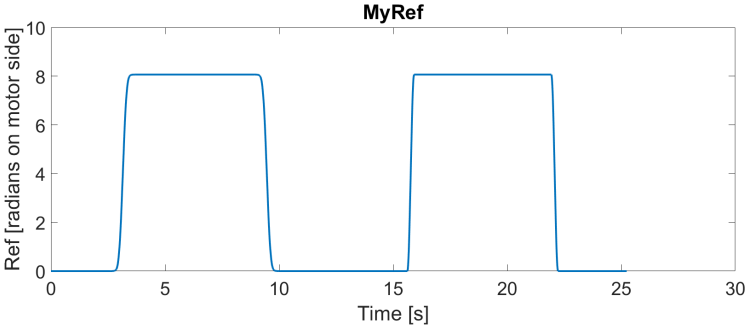


Figure 4.10 The reference values sent to the robot, first wave with filter, second one without.

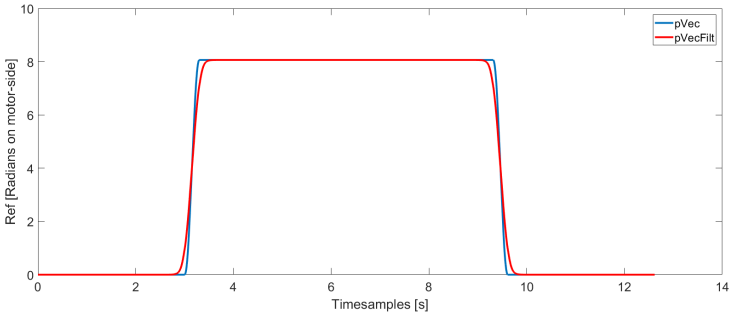


Figure 4.11 The reference values sent to the robot, plotted on top of each other.

Results from Swedish Robot

A notch filter was implemented on the reference to the position and velocity to see whether the calculated frequency was correct. The reference values sent to the robot can be seen below in Figure 4.10

with a comparison of the plots seen in Figure 4.11.

The resulting accelerometer values can be seen below in Figure 4.12.

The same movement was done twice. The movement that was filtered can be seen between samples 2000 to 4500, while the non-filtered movement can be seen between 5500 to the end. There is an obvious difference in excitement of the arm oscillations despite the small difference in shape of the two curves.

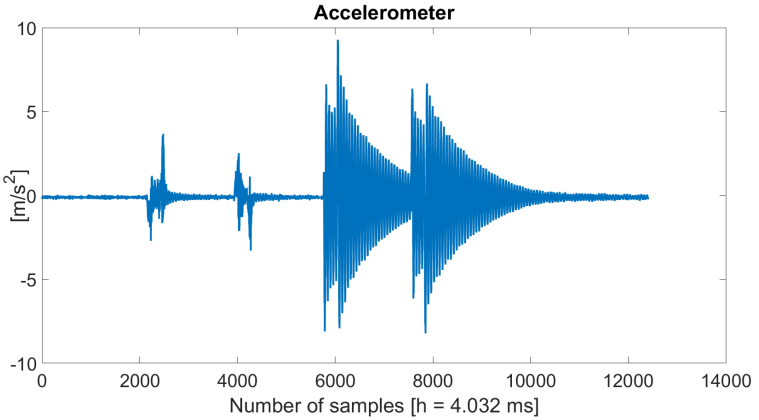


Figure 4.12 Accelerometer values of a movement, with and without notch filter.

5

Discussion

This thesis has been a long and arduous process of trying to work forward and decide what I should be doing. The original goal of this thesis was to research different control platforms and see how they differ in control structure. Slowly but surely, it has drifted from that point until it landed in the systems identification focused thesis you are reading right now. A good preparation would've been to take the course in Systems Identification[1], however, I did not at the time it was given, a mistake I have regretted.

Initially, I fell victim to the Dunning-Kruger effect and thought that the system I was going to test the different platforms on could be estimated using only a least-squares estimation. During the process of trying to estimate the ZP-4, I realized how wrong I was, and how deep the topic of identification goes.

Since I was uneducated in and on the field, many of the tests that were ran in Switzerland lacked the insight that was necessary to complete a proper estimation of the system. Almost all of them were run in closed loop, which can be seen to heavily influence the attempts to estimate the system.

It can be argued that the input signals were not exciting the system enough at the right frequencies, however, this is due to the physical limitations on the system. Running at the higher frequencies as seen in Figure 4.1 caused the cart, CMP1, to start to wobble, which is a non-intended behavior. Thus this was the higher limit of the system so as to not have the expensive machine break down.

Moving on to the estimation method analysis, the System Identification Toolbox was used heavily. Its methods were analyzed to see which would fit this specific problem the best. It was clear that a continuous grey-box estimation was most suitable. However, this might have been an effect of the symbolism in the discretization. Since discretizing a symbolic system is com-

[1] `http://control.lth.se/education/engineering-program/frtn35-system-identification/?L=0`

putationally heavy, the added load of trying to estimate a very complicated system matrix might've caused the estimation method to break down.

Attempts were made to estimate the system using the black-box method supplied by the toolbox. These were ultimately proven unsuitable, since not even the best estimation managed to replicate the output data given the same input data as the original system. The input data used in both grey- and black-box estimation was with the frictional component removed.

Now, this might be due to a limitation I unknowingly posed on my work. For the foundation-laying ground work, I tried to describe the simple system of a cart moving on a beam. For this, the friction component of the torque had to be described. Again, these tests were done in closed loop, which meant that rather than reaching a steady-state value, a lot of vibrations around the steady-state were seen. From what I could gather the loops in the drive used to control the system were not optimally tuned, but rather tuned too weakly. At higher speeds, this was less of a problem, causing the torque value to be slightly higher or lower. However, at slower speeds, this meant a lot of vibrations above and below zero value.

Another limitation I posed on myself was regarding the back-lash of the system. I decided not to try to describe this obviously real world phenomenon that exists in motors, since I decided it would require too much work, and play too little of a role in trying to control the system. However, it is important to still mention it, so as to disclose all possible sources of error.

Looking at the torque-velocity curve it nonetheless looks very close to how a "true" friction curve would look in an application such as this.

Last of things to discuss comes the base assumption, can a non-linear system be described well enough by a linear model? Unfortunately, I cannot say that I have gathered enough data to give a clear answer to this question. Looking at the Figures 4.4 & 4.5 there can definitely be discerned some movement towards the true system. However, the data that was used to estimate these systems has been found lacking, and further analysis of the system is needed.

I would like to close of this thesis by saying that I have learned an immense amount about systems identification, as well as approaching such a complicated subject in a structured manner. It is easy to get lost in details and not see the complete picture when trying to produce a result to be satisfied with, which can cause one to overlook faulty assumptions that have been made, such as: can a system be identified well in a closed loop.

6

Conclusions

Clearly no estimation of the system has reached a proper order of correctness. It seems that the sampling of the system was too quick originally for the estimators to handle, and such the downsampled signals should be used in further work. Since there is so low coherence between input and output of the system, further tests will need to be done in open-loop so as to acquire useful data. The Swiss system does seem to have a resonance located at around 6 Hz, which is good to know for where to look.

The best method of estimating such a system in a symbolical way thus seems to be to make a grey-box estimation of the system, using down-sampled signals and then discretize the system.

When looking at the outputs of the simulations using the estimated systems, one can see that the behaviour of the system can definitely be approximated, however, they do not seem to coincide at the same time, but rather one system dynamic is close to reality while the other drifts away.

This can be due to the estimators placing undue weights on one part of the outputs rather than all at the same time. Another can be that there are no "hard" constraints built in to the equations such as those in reality. CMP2 can in reality not drift too far off from CMP1, since the tension of the beam increases rapidly with each degree of bending. This is obviously one of the draw-backs of using a linear model to approximate a non-linear system.

The friction was estimated in a somewhat proper manner and would probably need further tweaking for improvement.

7

Further work

An obvious first step would be to re-do a lot of the tests in an open loop and see how the different measures of coherence hold up, and also the frictional model. Using this open-loop data, a strong estimation could be made, with which a control loop could be construed in such a way that no vibrations are induced when coming to a stop or starting from one.

Another point that my supervisors were interested in was the (transversal) vibrations in x-direction, i.e. in and out of the picture when looking at Picture 3.4. I made an assumption that this was due to the vibration in y-direction, however, this might not be the case.

Another interesting thing to look at might be how the vibrations of CMP2 depend on the extension in z-direction, i.e. how far down the weight has been lowered, and if it affects the system in a major way.

Bibliography

[1] L. Ljung, T. Glad, *Modeling and Identification of Dynamic Systems*, Lapaprint, Valmeria, Latvia, Edition 1:2, 2017

[2] K. J. Åström, B. Wittenmark, *Computer-Controlled Systems, Theory and Design*, Prentice-Hall, Inc., New Jersey, 1st Edition, 1984

[3] T. Glad, L. Ljung, *Control Theory - Multivariable and Nonlinear Methods*, Studentlitteratur, Lund, 1st Edition, 1997

[4] T. Hägglund *Reglerteknik AK - Föreläsningar*, Studentlitteratur, Lund, 1998

[5] V. van Geffen *A study of friction models and friction compensation*, DCT 2009.118, Technische Universiteit Eindhoven, Department Mechanical Engineering, Dynamics and Control Technology Group, Eindhoven, Netherlands, December 2009

[6] H. Olsson *Control Systems with Friction*, Department of Automatic Control, Lund University PhD thesis, TFRT-1045, Lund, Sweden, 1996

[7] S. C. Dutta Roy, B. Kumar, S.B. Jain, *FIR Notch Filter Design - A Review*, Electronics and Energetics vol. 14, No. 3, Serbia, 2001

[8] L.A. Aguirre, *The historical development of texts for teaching classical control of linear systems*, Annual Reviews in Control 39 (2015) 1–11, 2015

[9] R. Henriksson, *Observatörer för skattning av verktygspositionen hos en industrirobot*, Master's thesis report, LITH-ISY-EX–09/4271–SE Tekniska Högskolan i Linköping, Linköping, 2009

[10] E. Wernholt, *Multivariable Frequency-domain Identification of Industrial Robots*, PhD thesis, No. 1138, Linköping University, Sweden, 2007

[11] F. Bagge-Carlson, *Modeling and Estimation Topics in Robotics*, Department of Automatic Control, Lund University, Lic. Tech. Thesis, TFRT-3272, Lund, Sweden, 2017

[12] R. Johansson, *System Modeling and Identification* Prentice Hall. Inc, Englewood Cliffs, NJ, 1993

[13] R. E. Kalman, *A New Approach to Linear Filtering and Prediction Problems*, Transactions of the ASME–Journal of Basic Engineering, Volume 82, Series D, Pages 35-45, 1960

Lund University
Department of Automatic Control
Box 118
SE-221 00 Lund Sweden

Document name
MASTER'S THESIS

Date of issue
December 2018

Document Number
TFRT-6075

Author(s)
Marcus Peterson

Supervisor
Olof Sörnmo, Cognibotics
Anders Robertsson, Dept. of Automatic Control, Lund University, Sweden
Rolf Johansson, Dept. of Automatic Control, Lund University, Sweden (examiner)

Title and subtitle
Vibration Reduction in a Gantry Robot

Abstract

This report considers the problems of using a model-based approach to develop an estimation for a non-linear system using a linear model. Different estimation approaches were researched and evaluated using an ideal system, thus finding the suitable method for this thesis. The investigations considered both drive-train friction and the identification of oscillatory modes in a real industrial robot system of gantry-type and in a couple of laboratory setups, with different levels of achieved accuracy and reproducibility. A simple experiment with model-based reference compensation from identified oscillatory dynamics motivates the estimation problem and illustrate achievable benefits.

Keywords

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title
0280-5316

ISBN

Language
English

Number of pages
1-64

Recipient's notes

Security classification